

2005

PCT/CN03/00543

证 明

REC'D 27 AUG 2003

WIPO PCT

本证明之附件是向本局提交的下列专利申请副本

申 请 日： 2002 07 09

申 请 号： 02 1 23948.7

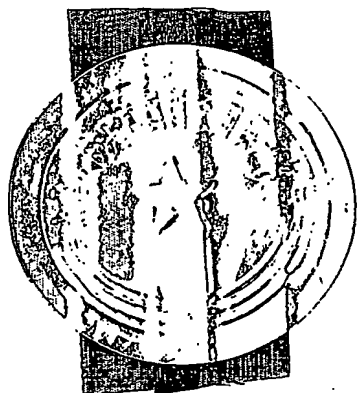
申 请 类 别： 发明

发明创造名称： 子序网模块及其调用

申 请 人： 徐肇昌

发明人或设计人： 徐肇昌

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)



中华人民共和国
国家知识产权局局长

王 景 川

2003 年 7 月 21 日

权利要求书

PRE/CN02/2270/SGL

1. 一种子序网, 所述子序网包括由 $N+1$ 个程序和分布的数据令牌结构组成的序网, 以及 $N+1$ 个条返回指令。
2. 根据权利要求 1 所述的子序网, 其中, 所述的子序网的调用指令是单机指令。
3. 根据权利要求 2 所述的子序网, 其中, 所述调用指令通过激发一个调用子序网部件实现对所述子序网的调用。
4. 根据权利要求 3 所述的子序网, 其中, 所述调用子序网部件包括一个调用总线部件, 用于完成从单机的调用位置到所述子序网的连接。
5. 根据权利要求 4 所述的子序网, 其中所述的调用子序网部件由一个单机序网调用指令启动, 该单机序网调用指令的调用参数被广播到 $N+1$ 个单元, 作为所述子序网的 $N+1$ 个分支程序的调用入口地址, 所述 $N+1$ 个分支程序进入时各有现场保护, 所述各分支程序以返回指令退出。
6. 根据权利要求 4 所述的子序网, 其中所述的调用子序网部件包括一个从串行单机的序网调用指令到 $N+1$ 并行的启动部件。
7. 根据权利要求 4 所述的子序网, 其中所述的调用子序网部件包括一个处理同时出现多个申请的调用竞争部件。

8. 根据权利要求 4 所述的子序网, 其中所述的调用子序网部件包括一个从串行到并行的调用参数传送和并行的中断启动部件。

9. 一种子序网调用系统, 包括:

多个单元计算机;

多个序网调用部件, 用于处理调用竞争, 保存序网调用参数, 和识别序网调用指令; 以及

一组总线, 所述一组总线将所述多个序网调用部件连接起来, 所述一组总线包括一个用于接收来自所述序网调用部件之一发出的启动电平的调用启动总线, 一组数据总线, 和一根共享的时钟总线。

10. 根据权利要求 9 所述的子序网调用系统, 其中所述的每个序网调用部件包括:

调用申请部件, 用于接受序网调用指令, 并在下一个时钟节拍产生一个启动电平送所述启动总线;

调用竞争部件, 用于仲裁一个优先权最高的调用有效; 以及

调用参数和中断部件, 用于根据所述调用竞争部件的输出确定序网调用参数的源, 并通过所述总线发出中断电平。

11. 根据权利要求 10 所述的子序网调用系统, 其中所述的每个序网调用部件还包括一个寄存和比较部件, 用于存储本机的权并将本机权与最高权相比较, 以便确定本机是否为序网调用参数的源。

- 7
12. 根据权利要求 11 所述的子序网调用系统,其中所述的寄存和比较部件还包括一个系统级寄存器和一个申请级寄存器,当所述序网调用部件接收一个级调用申请时,如果所述申请级寄存器的输出大于所述系统级寄存器的输出,所述的寄存和比较部件产生一个许可信号到所述调用申请部件,从而使所述调用申请部件产生送至所述启动总线的启动电平;而在所述申请级寄存器的输出小于所述系统级寄存器的输出时,所述的寄存和比较部件不产生所述许可信号,进而不产生启动电平。
13. 根据权利要求 10 所述的子序网调用系统,其中所述的寄存和比较部件还包括一个指示下一个序调用权的序调用寄存器,当所述序网调用部件接收一个序调用申请时,将所述序调用寄存器的输出和所述本机权进行比较,当比较结果相同时,产生启动电平;当比较结果不相同,不产生启动电平。
14. 一种包括由 $N+1$ 个程序和分布的数据令牌结构组成的序网以及 $N+1$ 个条返回指令的子序网的调用方法,其中,所述子序网包括多个计算机和多个序网调用部件,调用指令通过激发所述多个序网调用部件中的至少一个序网调用部件实现对所述子序网的调用,所述调用方法包括以下步骤:
- a) 所述多个序网调用部件接收来自所述多个计算机的序网调用指令;
 - b) 所述接收序网调用指令的序网调用部件向一个启动总线发出启动电平,送到所述全部序网调用部件;
 - c) 当所述接收序网调用指令的序网调用部件在测试到所述启动总线有所述启动电平时,所述接收序网调用指令的序网调用部件控制向所述数据总线送出一个本机

权；然后所述数据总线上全局的权被所述的序网调用部件写入一个暂存器；以及

- d) 所述接收序网调用指令的序网调用部件对所述暂存器的最高位和所述本机权比较；当比较结果相同时，所述序网调用部件的调用有效；当比较结果不同时，所述序网调用部件的调用无效；至少一个接收序网调用指令的序网调用部件的调用有效。
15. 根据权利要求 14 所述的调用方法，其中所述的步骤 b) 还包括将所述接收序网调用指令的序网调用部件发出的本机权在或数据总线上进行“或”运算产生所述全局的权的步骤。
16. 根据权利要求 14 所述的调用方法，还包括在步骤 a) 和步骤 b) 之间由所述多个计算机中的至少一个计算机发出一个用于支持重要性不同的有等级的级调用指令的步骤。
17. 根据权利要求 16 所述的调用方法，还包括先于所述的计算机发出所述级调用指令，所述的计算机将所述调用子序网的申请级写入所述序网调用部件的申请级寄存器的步骤。
18. 根据权利要求 17 所述的调用方法，还包括通过对所述申请级和系统级的最高位进行比较的步骤，其中，若所述申请级低于或等于当前所述系统级，则所述级调用申请不被接受；若所述申请级高于所述当前系统级，则所述级调用申请被接受，并且该序网调用部件在步骤 b) 发出启动电平。

9

19. 根据权利要求 18 所述的调用方法, 还包括在步骤 b) 和步骤 c) 之间的以下步骤:

b₁) 在所述启动总线上对来自所述接收到序网级调用指令的序网调用部件的所述启动电平进行或运算并将计算结果传播到所述的全部序网调用部件;

b₂) 所述的接收到序网级调用指令的序网调用部件在测试到所述启动总线有所述启动电平时, 接收到序网级调用指令的所述序网调用部件控制向所述数据总线送出一个本机的申请级, 然后所述数据总线上全局的申请级被所述的全部序网调用部件各自写入一个暂存器;

b₃) 所述接收到序网级调用指令的序网调用部件将暂存器的最高位和申请级进行比较, 若比较结果相同, 表示该序网调用部件的级调用有效; 以及

b₄) 将所述暂存器中的最高位写入所述全部序网调用部件的系统级寄存器, 以记录系统新的最高级。

20. 根据权利要求 14 所述的调用方法, 还包括在步骤 a) 和步骤 b) 之间由所述多个计算机中的至少一个计算机发出一个所述序网的序调用申请用于支持程序处理的有序事件和复杂的程序结构的步骤。

21. 根据权利要求 20 所述的调用方法, 还包括在全部序网调用部件的寄存器中指定下一个序调用机号的步骤。

22. 根据权利要求 21 所述的调用方法, 还包括在步骤 b) 和步骤 c) 之间的以下步骤:

b') 所述多个序网调用部件接收来自所述计算机的序调用申请指令; 以及

b'') 将所述序调用机号和本机号进行比较，若所述序调用机号和所述本机号相等，则所述序调用申请被许可，送所述启动电平到所述启动总线；若本机号和所述序调用机号不同，或所述序调用机号内容无效，则序网调用部件的序调用申请无效，无所述启动电平送至所述启动总线。

23. 根据权利要求 22 所述的调用方法，还包括在步骤 c) 和步骤 d) 之间的以下步骤：

- c') 由接收到序网序调用指令的序网调用部件送序调用特征数据到数据总线；以及
- c'') 所述接收到序网序调用指令的序网调用部件将所述数据总线上的内容写入暂存器并进行序调用特征检查以判断没有级调用存在，即所述序调用有效。

子序网模块及其调用

本发明所属技术领域

本发明涉及多计算机技术，尤其涉及序网计算机技术。

与本发明相关的背景技术

面对长期不能够获得突破的并行处理技术，先有的序网计算机的发明(中国专利申请号 00119472.0)已经找出了 $N+1$ 个流和 $N+1$ 个序的数目匹配问题。并说明了流与序的数目匹配是通用性的必要条件。同时，根据通用性目标，先有序网计算机发明创建了有 $N+1$ 个程序的序网计算机结构。序网计算机支持没有参量约束的 $N+1$ 程序的序网，实现了通用性的第一步。

本发明进一步发展序网计算机的通用性。它将序网的并行程序的可封装特性进一步转为可以调用、自动返回的子序网并行模块，简称子序网。并用子序网调用结构建立多序网的组合形式，拓展了序网计算机结构 and 应用模型的适配能力。

本发明在图 1 中建立起单机的调用子程序部件结构。在图 2 中介绍了先有技术序网结构并建立子序网。从图 3 开始进入调用子序网的结构描述。

图 1A 是单机中的子程序调用。“调用和返回”是计算机的二条指令，是处理程序嵌套结构的元件。在图 1A 中，程序 101 中的一条调用指令 100，可以实现一个子程序 102 的调用。调用指令 100

嵌在原程序 101 中，含有子程序 102 的入口地址参数。执行调用指令 100 后，程序进入子程序 102，子程序 102 中的返回指令执行后，程序返回到原程序 101 继续运行。

调用指令 100 可以分解为调用位置和调用参数二个参量。调用位置是指令在原程序中的位置，它与原程序中的其他指令有序的关系，而与子程序无关。调用参数指示子程序入口地址，与原程序结构无关。子程序模块包括入口地址、程序体、和返回指令三个参量。其中“程序体+返回指令”组成了子程序，因此，“调用指令+程序体+返回指令”过程称为“调用子程序”。支持调用子程序的硬件结构称为调用子程序部件。

调用子程序部件的实际结构是：单机程序运行到调用指令，产生调用位置和调用参数的操作。位置的操作是将指令计数器的内容推入栈区，调用参数的操作是将参数送指令计数器。然后，下一条指令以参数作为地址去执行。子程序结束时，执行一条返回指令，将栈内的数据弹出，送到指令计数器，使下一条指令按弹出的地址执行。这个过程相当于原程序返回并继续执行。显而易见，在单机 CPU 的结构中，调用子程序部件的结构是分散的、有些结构是和其他指令共享的。本发明是将这些结构组合起来，定义成一个独特的调用子程序部件。

调用子程序的外部特征是调用位置和子程序模块，调用指令和被调用程序的连接（即调用参数和子程序入口地址）成为内部连接。

从原程序角度来看，一条调用指令可以激活调用子程序部件，调用指令为这个元件提供了位置和参数二个参量。调用结构子程序元件根据位置保护程序地址，根据调用参数指定子程序入口。当子程序执行返回指令时，调用子程序部件处理程序返回原来的位置。

调用子程序部件中的被调用程序和返回指令采用了单机中通用的程序和指令硬件结构，因此，调用子程序部件的特有部分是从调用指令开始--参数传递---子程序启动阶段的硬件结构

调用子程序部件内的信息关系是：调用指令中的“调用位置”启动调用子程序部件；“调用参数”在调用子程序部件内部传送，指示和子程序模块连接，“子程序”内含返回指令，使返回原程序。

“调用位置--调用参数--子程序”是调用子程序部件内的信息处理过程。

图 1B 和图 1C 反映了调用位置、调用参数、和子程序的关系。从信息角度来看单机调用子程序部件，图 1A 表示了嵌套关系，图 1B 表示在不同调用参数情况下，同一个调用位置 **108** 可以实现不同的子程序调用；图 1C 表示在相同调用参数情况下，不同的调用位置 **108** 可实现同一个子程序调用。

图 1D 是基于调用信息来描述的调用子程序结构示意图。在单机环境下，调用位置、调用参数和子程序模块可以形成了如图 1D 所示的结构特征。其中 **108** 是调用位置，**102** 是子程序模块，**104** 是一条虚拟的总线结构。虚拟总线 **104** 的功能就是调用参数传送，实现位置和子程序的连接。

调用位置 **108** 嵌套在程序 **105** 中。图 1D 清晰地显示了调用位置 **108** 的集合和子程序 **102** 的集合之间的结构分层关系。当然，这是程序结构上的分层，在时间上是串行的。其中，程序 **105** 含调用指令，激活调用子程序部件，启动子程序 **102** 集合中的一个子程序。调用位置 **108** 的集合和程序 **102** 的集合组成了 Client—Server 结构，表示调用子程序部件是支持单机程序的 Client—Server 结构的核心。

图 1D 是从信息角度对调用子程序部件的一个描述方法。它突

14
出了调用参数的传送--虚拟总线的连接结构。它的实际功能和图 1A 一样。然而，在多机环境下，信息必定有外部连接，因此和图 1D 的单机表达方法有相似之处。

调用子程序部件是一个共享的硬件结构，必须时分使用。调用子程序部件运行过程包含有三个环节：

- 调用位置—调用参数—子程序模块

其中，调用位置启动调用子程序部件；调用参数是被传递的；子程序模块是运行结构。启动时程序有现场保护，在子程序模块中有返回指令。

图 1E 是现有技术中常见的硬件中断。中断信号 106 连接计算机 CPU107 的中断引脚，使计算机 107 能转入中断程序。

如图 1F 所示，中断对程序结构的影响是：使原有的程序 101 停顿，保护停顿位置并进入中断程序 109，中断程序 109 结束有中断返回指令，继续原有程序 101 的运行。显见，中断结构和调用指令有很多相似之处。其中，中断矢量相当于调用参数，中断程序相当于子程序。中断和调用的区别在于中断位置和被中断程序没有关系，而调用位置和原程序上下的事件有序的关系。

中断结构运行过程包含有三个环节：

- 中断位置—外部启动中断的硬件结构，确定程序的中断位置。

- 中断矢量—指示中断程序入口。

- 中断程序进入—中断矢量的程序，并利用中断返回指

令返回原程序。

从计算机硬件来看，除了进入调用子程序部件的前端处理方式不同外，中断和调用指令的特征基本一样，都有程序嵌套的效果。这一事实表明，调用指令和中断可以共享（大部分）调用子程序部件。而在进入调用子程序部件前的处理方式上，二者的区别已经被分别用于处理已知的有序事件和外部的随机事件。因此，也可以这样说，“有序事件和随机事件共享单机的调用子程序部件”。

图 2A 是序网及封装示意图。先有发明序网计算机是一种分布结构的并程序（中国专利申请号 00119472.0）。如图 2A 所示，它具有分布的数据令牌结构 204 和 $N+1$ 个（图中 $N=3$ ）程序流 210Ps, 211P1, 212P2, 213P3。其中，每一个分布数据令牌 204 包含有一个一致化令牌指令 201、一个源数据令牌指令 202 和多个目的数据令牌指令 203。分布数据令牌 204 在内部进行数据和令牌的一致化操作。程序 P1—P3 由源数据令牌指令 202 和目的数据令牌 203 组成。一致化程序 Ps 全部由一致化令牌指令 201 组成。各程序对数据令牌指令的读写操作是限制在局部（或本地的）进行的。

序网是由 $N+1$ 个程序的独立驱动运行所完成，程序运行的原理如下：

1. 在遇到特定的写数据指令时，程序向本地的数据令牌写入数据、产生源令牌、然后程序继续运行。如此，构成了序网的源数据令牌。

2. 在遇到特定的读数据指令时，程序读本地的数据令牌的数据、由硬件实现对同地址令牌的测试。若测试结果有效，程序继续运行；若测试结果无效，则该程序踏步，以等待有效的数据到来。如此，构成序网的目的数据令牌。

3. 在遇到一致化指令时，执行该分布数据令牌的数据令牌一致化操作，一致化操作是数据和令牌从源到目的的传送。一致化指令测试一致化操作（传送）中的令牌值。若测试结果无效，则一致化指令自动重复，直至有效的（数据）令牌到来；若测试结果有效，则一致化指令结束，程序继续运行。如此，构成序网的一致化令牌。

序网的封装

根据序网的定义，多程序间的互连和同步等待都能在序网内部自行解决。因此，只要序网的 $N+1$ 个程序能进入运行，序网自己就能完成序网的计算，无需外界的帮助。因此，序网可以以 $N+1$ 个程序段方式来实现封装。

序网的可中断特性

从序网的工作原理来看，序网是一个并行程序。任何一个分支程序发生暂停，都会使序网进入停顿状态。但不会使序网出错。而一旦该程序恢复运行，则序网可以继续运行。因此，序网的每一个分支都允许被中断，而在 $N+1$ 个分支同时被中断的情况下，如果 $N+1$ 个中断程序组成另一个序网，表示序网被另一个序网中断。

因此，序网的二个重要应用特征是：

- 可封装性：启动 $N+1$ 个序网分支运行，序网自行完成并行模块的功能。

- 可中断性：任何一个运行中的序网允许被另一个序网中断。中断不影响序网运行的正确性。

在先有的序网计算机发明中已经表明，过去大部分可运行的并行处理计算机实际上只能支持一个序网。它对变化多端的应用模型

显然是不适应的，限制了并行计算机的应用范围。本发明建立的序网计算机开始支持多序网，多序网之间的连接采用调用子序网的结构。

本发明的目的

本发明的一个目的是创建子序网。子序网是序网封装的一个特殊结构。它的结构特点是由 $N+1$ 个单机子程序组成，从而，使调用子序网的结构可以借用单机的子程序结构。

本发明的另一个目的是创建序网计算机环境下的序网调用功能。本发明首次创建序网调用的功能。它创建一种新的序网调用指令。序网调用指令是单机指令，可以调用一个子序网 ($N+1$ 个分支) 的运行。

用序网调用指令激发“调用子序网部件”，实现序网的调用。调用子序网部件保持了“调用位置-调用参数-子序网”三个过程环节。它由任何一个单机序网调用指令启动，该序网调用指令的调用参数被广播到 $N+1$ 个单元，作为序网的 $N+1$ 个分支程序的调用入口地址， $N+1$ 个分支程序进入时各有现场保护，各分支程序以返回指令退出。

本发明的目的是创建序网的级调用和序调用二种申请模式。序网计算机通用性的二个特征是序网和序网的调用。为了能适配模型中的有序事件和随机事件，本发明还创建了序网的级调用和序网的序调用二种调用申请模式。

附图的简要描述

下面，本文进入本发明的结构描述，并在所用的名词上，作了统一。各图中的相同序号和字母代表相同的结构或意义。本发明的

新颖独特的特征，使用方法，发明目的等内容在参照以下将要详细描述的实施例和相关附图后，会得到很好的理解。其中：

图 1A 是单机中的调用子程序结构特征；

图 1B 是不同调用参数调用不同子程序的示意图；

图 1C 是多个调用位置调用同一个子程序的示意图；

图 1D 是基于调用信息来描述的调用子程序结构示意图；

图 1E 是硬件中断示意图；

图 1F 是中断操作的程序结构示意图；

图 2A 是序网及封装示意图；

图 2B 子序网结构示意图；

图 3A 是调用子序网部件结构图；

图 3B 是中断调用序网的硬件结构示意图

图 3C 是中断调用序网的程序结构示意图；

图 3D 是调用子序网部件和程序关系图；

图 3E 是调用子序网部件的步骤流程图；

图 4A 是序网调用系统 SNCS 的结构图；

图 4B 是序网调用部件 SNCD 的结构图；

图 4C 是序网调用部件 SNCD 的工作流程表;

图 4D 是 CAPD 部件的合作控制;

图 4E 是 CMPD 部件的合作控制;

图 4F 是 C&ID 部件的合作控制;

图 4G 是 R&CD 部件中的比较部件结构;

图 5A 是本发明的序网级调用结构;

图 5B 是本发明序网级调用工作流程表;

图 5C 是本发明的序网序调用结构; 以及

图 5D 是本发明序网序调用工作流程表。

本发明的详细描述

图 2B 是子序网结构示意图。序网封装是并行程序的封装, 如果序网封装中的每一个序网分支程序用单机的返回指令 205 结束, 该序网封装称为子序网。子序网是一种特殊的序网封装形式。由于序网的分支程序和子程序相当, 因此, 子序网在结构上相当于 $N+1$ 个子程序的组合。其特征是:

- $N+1$ 个程序的入口地址。
- 每一个分支用返回指令结束 (表示进入时, 需要地址保护)

根据序网运行的特征, 只要在子序网进入时, $N+1$ 个分支程序

有现场保护,则在序网分支程序结束时,就可以简单地用 $N+1$ 个独立的单机返回指令,来恢复调用前的序网运行。这表明子序网的返回结构可以借用 $N+1$ 个单机的调用子程序部件结构。子序网的嵌套直接是 $N+1$ 个分支程序的嵌套形式。

图 3A 是调用子序网部件结构图。这是在多机环境下,通过序网调用指令调用子序网的结构图。它采用类似于图 1D 的基于信息的表示方法。包括调用位置--调用参数--子序网三个环节。

序网调用指令是单机指令格式,包含调用位置 300 和调用参数。在程序运行到序网调用指令时,产生序网调用位置和序网调用参数。调用位置 300 是单机程序中的位置。每一个调用参数对应一个子序网 302。调用参数是调用位置 300 和子序网 302 之间的连接,构成了调用总线部件 304。调用总线部件 304 完成从单机的调用位置到 $N+1$ 个子序网分支的连接。实现调用位置--调用参数--子序网的结构,称为“调用子序网部件”。

从发动调用的原程序角度来看,一条单机的序网调用指令可以激活调用子序网部件,调用指令要为 $N+1$ 个分支程序提供了调用位置和调用参数二个参量。而各分支程序在有了位置和调用参数二个参量后,各单元计算机实际上是在单机的调用子程序部件支持下,执行一个子程序的调用和子程序返回。

在多机环境下,上述“调用指令要为 $N+1$ 个分支程序提供了调用位置和调用参数二个参量”是通过调用总线部件 304 来完成的。调用总线部件 304 不是简单的连接结构,它要完成调用参数的广播,和启动 $N+1$ 个序网分支程序。

调用总线部件 304 的输入来自程序的序网调用指令 300,输出是序网 $N+1$ 个分支程序的运行。因此,调用总线部件 304 有一个单

机变并行的过程。同时，序网调用指令是单机指令，可能会有多个序网调用指令同时出现。而调用子序网是全局结构，一次只能处理一个。因此，调用总线部件 304 还要有处理多个序网调用指令的竞争结构。

由于存在竞争，调用总线部件 304 的结构必须可以同时包括 $N+1$ 个调用位置 300 的输入，如图 3A 所示，和它们的竞争处理。相比之下，单机的串型结构自动消除了调用上的竞争，图 1D 所示的多调用指令 108，不会同时产生，没有竞争。

和单机的调用子程序相比，多机的调用子序网也由（单机）调用指令发动；在调用位置和参数传送过程中，有串型到并行的展开；和多个调用同时发生的竞争处理；而子序网是单机子程序的组合（其中序网互连数据在子序网内部解决，和调用结构无关）。

序网调用指令需要新的硬件结构设计。序网调用指令仍保持二个参量，即序网调用位置 300 和序网调用参数。而在用户界面上，仍是传统的调用 CALL 和返回的概念。

调用子序网部件的工作过程如下：

- 程序运行到序网调用位置 300。经过 303，将调用位置 300 和调用参数，送调用总线部件 304。这是各单元的本地操作。

- 调用总线部件 304 可能接收到多个序网调用位置 300 产生的启动电平，通过部件 304 内部竞争，仲裁一个序网调用有效，屏蔽其余的序网调用。

- 调用总线部件 304 经 305，将有效的序网调用参数广播传送到各个分布单元。一个序网调用参数和一个子序网 302

对应。

- 进入子序网 302 的各个序网分支，有现场保护。
- 子序网各分支程序有返回指令，退出序网。

调用子序网部件运行过程有三个环节：

- 调用位置-调用参数-子序网

调用位置启动调用子序网部件；调用参数是在调用子序网部件中传递的内容；子序网有并行运行的结构。子序网各分支程序在启动时有现场保护，用返回指令结束。其中，调用子序网部件结构还包括：（1）处理调用竞争的硬件环节。（2）从单机到多机的调用参数传送结构。

调用子序网部件中的调用位置和单机调用位置的不同点的说明如下：

在多台系统中，一个调用位置应该有二个定位参数。即“机号”和“程序中位置”。由于不同机号可以由调用-序网元件的竞争规则解决，使用户编程中的调用子序网指令不需标明机号，这对编程是有利的。对竞争失败的序网调用指令，系统可以建立统一的重新申请处理，不需要用户编程。

图 3B 是中断调用子序网的硬件结构示意图。调用子序网操作的最后是启动子序网的分支程序。如本发明图 3A 所描述，需要在启动时对现在运行的程序进行保护。本发明采用了单机的中断结构，可以方便地保护被中断程序，也有利于和单机结构的兼容。如此，使调用子序网部件出现“调用位置产生 $N+1$ 个中断”，和“ $N+1$ 个中断产生序网”的分段路径。

总线 306 连接各计算机中断，总线 306 上的一个中断信号同时对各计算机发出中断请求，产生 $N+1$ 个计算机程序 107 的中断。当这些中断矢量代表同一个序网的各个分支时， $N+1$ 个中断程序实现了子序网的调用。

图 3C 是中断调用子序网分支程序结构示意图。 $N+1$ 个计算机在运行 $N+1$ 个程序，本图中用 310、311、312、313 来表示。在接受了来自中断线 306 的信号后，各自产生了中断程序 320、321、322、323。由于程序 320、321、322、323 组成了一个序网，根据序网可以自行完成运行的原理，并行进入中断程序就实现了序网的调用。在中断发生时，原来的程序 310、311、312、313 各自保留中断时的程序状态。序网各个分支程序结束时，返回指令分别恢复了原有的 $N+1$ 个程序 310、311、312、313 继续运行。图中 Ps、P1、P2、P3 用同一个符号表示他们实际上是同一个计算机程序。

在实用中，计算机的中断矢量本来就是紧缺资源。在应用中，子序网又需要很大的编号空间，不可能由中断矢量空间来提供。因此，从中断到序网执行过程需要改进。本发明采用方法如下：中断前，调用参数在调用总线部件 304 内广播，使个单机中断程序都可以从调用总线部件 304 中读取该调用参数；中断总线 306 连接固定的中断矢量，进入固定的中断程序；由中断程序读取调用参数，并转入调用参数指示的序网入口地址。当 $N+1$ 个中断程序都这样执行后，系统进入了序网运行。

在利用并发中断启动子序网运行后，调用子序网部件还需要一个能连接单机调用指令和并发中断的硬件结构 304。经过该硬件结构，一条序网调用指令可以转为 $N+1$ 个程序的中断。

图 3D 是调用子序网部件和程序关系图。本图是图 3A 结构功能的另一种描述方法。在本图中，将调用子序网部件和序网调用指令、

程序一起来描述，结构就比较形象。调用子序网部件由硬件和软件共同组成。

从序网调用指令到子序网分支程序的运行过程分成六个阶段，在图中用罗马数字表示。它们是 I 调用申请、II 调用启动、III 调用竞争、IV 调用参数和中断、V 产生序网入口、和 VI 序网运行。本图通过各阶段的 N+1 个程序的状态描述，清楚地表明了本发明调用子序网部件的结构特征。其中，有同一标号 310, 311, 312, 313 的二段程序分别表示调用申请阶段和接受中断阶段的状态。相同的 Ps、P1、P2、P3 标号表示他们是同一个计算机中的程序。

1. 序网调用的第一个阶段是调用申请阶段。序网调用指令是单机指令格式。在多机系统环境下，程序 310-313 中的任何程序都可能出现序网调用指令来启动调用子序网部件。如图所示，在程序 311 中的序网调用指令 319，和程序 313 中的序网调用指令 318 正好同时产生启动电平。进入调用总线部件 304 的处理过程。单机本来没有序网调用指令，需要指令扩充。单机指令的扩充方式很多，在 CPU 中直接设计最好。本发明实施例利用 IO 指令改造。改造方式是指定 IO 地址为序网调用指令，由 IO 设备的硬件结构解释。例如，可以将写序网调用参数的 IO 指令同时定义为序网调用指令。因而，该指令在写序网调用参数时，产生调用启动。
2. 序网调用的第二阶段是调用启动阶段。序网调用指令中的序网调用位置，经 IO 设备的硬件译码，产生一个启动电平。该启动电平经过调用启动总线 402，传送到分布结构的调用总线部件 304。如果同时出现多个调用启动，它们在调用启动总线 402 上合并成一个。调用启动总线实现了 1 到 N+1 的数目扩展。调用启动阶段占用一个时钟节拍 H0。

3. 序网调用的第三阶段是调用竞争阶段。多个启动电平在同一时刻产生（如图中 318, 319）时，产生调用竞争。调用竞争在调用总线部件 304 中实现。调用竞争阶段仲裁出一个调用有效。调用竞争在分布结构环境下完成，包括信息交换、比较和决策等操作，占用一个或多个节拍。调用竞争需要一个优先权规则，规则在调用总线部件 304 内部用硬件来执行。本发明实施例采用本机权（本机机号）数字小为优先的简单排列规则。图中，调用总线部件 304 的竞争仲裁确定序网调用 319 为有效，也同时确定了本次序网调用的调用参数 317（来自程序 P1）。调用竞争阶段占用一个时钟节拍 H1。
4. 序网调用的第四阶段是调用参数和硬件中断阶段。调用竞争阶段确定了一个序网调用参数 317。在本阶段，调用参数 317 在调用总线部件 304 内广播，并分布地存储在各中断程序的可读端口，用于指示序网的入口地址。同时，在一个时钟管理下，调用总线部件 304 向各个计算机发出硬件中断信号 316。程序 Ps、P1、P2、P3 各自被中断，进入指定的中断程序。调用参数和硬件中断阶段占用一个时钟节拍 Hint。
5. 序网调用的第五阶段是产生序网入口阶段。各中断程序 320、321、322、323 在完成中断保护处理后，从调用总线部件 304 读入调用参数。中断程序解释调用参数，并各自使程序执行中断保护，转入由调用参数指示的、序网的分支程序。本阶段是软件过程，在图中和第六阶段合并。
6. 序网调用的第六阶段是序网执行和返回阶段。序网运行只要求 N+1 个程序进入运行。序网各分支程序 320、321、322、323 运行结束有返回指令，使各单元的计算机继续原程序 310、311、312、313。

26

本发明序网调用选用中断结构的原因是中断有地址保护，使调用序网的返回可以用 $N+1$ 个中断返回指令的简单处理方式。本发明选用中断的原因是因为中断是单机系统的开放资源，也易于实现硬件并发。同时，又可以保持单机资源的完整性，以达到整机结构进入新系统的兼容模式。

在本发明的调用竞争阶段，是根据各机的权来处理优先级。各计算机中有一个本机权寄存器，其内容被预先设置（不能有权相同的预设置情况）。本机权在运行过程中一般是不变动的。各计算机中还有比较部件结构，可以找出本机权最高的计算机。根据需要，本机权可以和本机的机号代替。

子序网分支程序保护调用位置的一个特殊情况说明如下：调用总线部件 304 和中断连接完成了序网调用位置从 1 到 $N+1$ 的展开和中断程序的位置保护。但是对发动调用的程序来讲，出现了一个特殊情况。即序网调用指令的调用位置和调用总线部件 304 的中断位置在时间上是二个点。当然，调用位置的二个时间点发动调用程序的结构上必须合并成一个点。用硬件来实现，意味着中断必须在序网调用指令结束前产生；用软件来实现，需要在序网调用指令之后，要有一个软件延时检测环节。由于做到这个要求并不难。在下面的描述中，本发明将不再作上述位置合并的说明。

图 3E 是调用子序网部件的步骤流程图。本图描述本发明调用子序网部件的六个阶段分割。

1. 程序运行到序网调用指令，分解出调用参数和调用位置二个参量。
2. 调用位置产生启动电平，进入调用启动总线。
3. 调用竞争过程，仲裁一个调用有效。

- 27
4. 确定有效的调用参数广播;同时有 $1+N$ 个并发中断电平产生。
 5. $N+1$ 个中断程序各自读取调用参数,解释后,转入序网的入口。
 6. 在 $N+1$ 个程序驱动下,实现序网,各分支用中断返回指令结束。

其中,1—5 是序网的调用过程,其输入是嵌入原程序的单机调用指令,输出是 $N+1$ 个序网的入口矢量。6 是序网的执行部分。

如图所示的另一个划分是:1 是单机的执行,2-6 是全局的并行执行。如图所示的另一个划分是:1-4 是调用子序网部件的硬件结构部分,源程序的单机调用位置开始,到广播调用参数和产生中断电平。5-6 是调用子序网部件的软件部分。在下面图 4 的实施例中,本发明描述一个调用子序网部件的硬件实施结构,其内容就只包括 1-4 的步骤。

图 4A 是序网调用系统 SNCS 的结构图。本图是图 3A 或图 3D 结构的一个硬件实施例。在图 3E 中的全程 1-6 阶段中,由于第 5-6 阶段进入了中断程序软件范围。本图的硬件结构设计从序网调用指令开始,到完成调用参数的广播(使中断程序可读),和产生硬件中断的电平为止。即图 3E 的 1-4 阶段。显然,它相当于图 3A 的调用位置 300 和调用总线部件 304。

为了适应超大规模集成电路技术的发展,本发明实施例的硬件必须有分布和同构的特征。为了有整个单机结构的兼容性,本发明实施例所占用的单机资源必须是单机结构的开放资源。

本实施例由 $N+1$ 个单元组成,每个单元包括一个计算机 400 和

一个序网调用部件 **SNCD 407**。其中，计算机 **400** 允许异构，**SNCD** 部件 **407** 必须是同构的。每一个 **SNCD** 部件作为计算机外部设备和计算机的 **IO 总线 405** 连接。全部 **SNCD** 部件之间通过一组总线互联，其中包括一根调用启动总线（简称启动总线）**402**，一组数据总线（本实施例中是 16 根）**404** 和一根共享的时钟线 **403**。通过共享的时钟线 **403** 全部 **SNCD** 部件 **407** 可以在一个时钟的节拍下合作运行。

全部的 **SNCD** 部件 **407** 和总线组成了序网调用系统 **SNCS406**。在某个计算机程序 **400** 执行序网调用指令后，**SNCS** 系统 **406** 进入运行，直至为全部计算机 **400** 准备好调用参数，和输出中断电平信号 **INT 408**。因此，序网调用系统 **SNCS406** 相当于图 3A 中的调用位置 **300** 和调用总线部件 **304**。如前所述，各个计算机在接收到中断信息后，各自转入中断程序。中断程序从各自匹配的 **SNCD** 部件 **407** 中读取序网调用参数，解释后转入序网运行。本实施例的序网调用系统 **SNCS406** 是硬件结构，不包括中断程序和中断返回指令等软件处理部分。

数据总线 **404** 和启动总线 **402** 都是“或”总线结构。或总线结构的特点是允许多机 **SNCD** 部件 **407** 同时向总线发送数据。或总线也是一种运算方式。例如，按每个 **SNCD** 部件一位的结构，16 位的数据总线 **404** 可以在一个节拍内生成一个包括 16 个计算机信息的全局信息。

序网调用系统 **SNCS406** 的结构是分布的，每一个 **SNCD** 部件 **407** 独立地和本地的计算机 **400** 接口。本实施例中，**SNCD** 部件 **407** 以外设方式和计算机连接，包括一组计算机的外设总线 **405**，和计算机的中断线 **INT408**。在外设结构下，总线 **405** 包括地址总线 **A**，数据总线 **D** 和控制总线 **C** 三部分。对 PC 机来讲，这组总线可以是 **ISA**、**PCI**。在非外设总线结构下，总线也可以是 PC 机的内存总线

29
等。

本实施例结构的工作原理是计算机 400 运行序网调用指令，经总线 405，向 SNCD 部件 407，发送一个序网调用参数和一个序网位置信号。该 SNCD 部件 407 发出一个启动电平，在启动总线 402 上的传播。全部 SNCD 部件 407 在检测到启动电平后，在同一时刻，各自进入 H0 节拍。在下一个 H1 节拍，各 SNCD 部件 407 一起进入调用竞争阶段。在调用竞争阶段仲裁出一个调用申请有效的 SNCD 部件 407。然后，在下一个 Hint 节拍中，调用有效 SNCD 部件 407 的序网调用参数在 SNCS 系统 406 内广播。各 SNCD 部件 407 读入和存放该调用参数，以供计算机中断程序的读取。同时，由各 SNCD 部件 407 产生中断电平 408 送各自的计算机 400，使计算机 400 进入中断程序。

本实施例的序网调用指令是单机指令格式。在采用计算机 IO 指令来实现序网调用指令时，需要占用 IO 地址。当计算机利用特定的 IO 地址进行写调用参数操作时，SNCD 部件除了保存序网调用参数外，还将 IO 地址识别为序网调用指令，然后向启动总线 402 发送启动电平。显见，序网调用指令不是传统的单机指令。它要用单机的 IO 读写指令，并通过一系列操作才能在用户编程界面表现一个传统的调用概念。

本发明描述的分布结构有 $N+1$ 个单元，这是序网运行的程序数目。对于调用结构而言，实际上没有程序数目的约束。在先有的序网发明专利中，第 $N+1$ 个程序在硬件上被设计成由 N 个分布和同构的 Cell 结构组成，由 N 个 Cell 合作产生第 $N+1$ 个程序。因此，实际有 $2N$ 个（程序）结构，实际的调用子序网部件也是同时处理 $2N$ 个（程序）中断的结构。本发明用 $N+1$ 个程序来描述的原因是： N 个 Cell 的程序是同步运行的，共享一个程序码；它们硬件的并发中断也是同步的，和一个中断的过程一样； N 个中断程序也仍是共享一个程序码，并同步运行。因此，实际过程和一个程序的中断是

一样的。

图 4B 是序网调用部件 SNCD 的结构图。SNCD 部件 407 内部结构包括四个部件,它们是调用申请部件 CAPD410,调用竞争部件 CMPD411,调用参数和中断部件 C&RD412,以及寄存器和比较部件 R&CD413.

R&CD 部件 413 内部有三个 16 位寄存器,它们是调用参数寄存器 VECT、本机权寄存器 LOCR 和暂寄存器 TMP。计算机 400 的 IO 指令可以读写寄存器。在部件 410、411 和 412 的控制下,寄存器也可以读写数据总线 404。在 R&CD 部件 413 内部,还有比较部件 COMP,比较结果经 415 送 CMPD 部件 411。

调用参数是一个 16 位 (或 15、14 位) 的数据,指示序网的入口地址,存放在调用参数寄存器。本机权寄存器 LOCR 中每个位 (=1) 表示 1 个权,共 16 个权。各部件的本机权设置不能有二个权相同的情况。TMP 是暂寄存器,在不同的时间存入不同的数据。

CAPD 部件 410、CMPD 部件 411、C&RD 部件 412 分别用于调用申请阶段、调用竞争阶段、调用参数和中断阶段的管理,三个阶段在时间上是连接的。

CAPD 部件 410 是本地操作,它经 405 接受来自计算机序网调用指令,并在下一个时钟节拍产生一个启动电平送启动总线 402。

同时,所有的 CAPD 部件 410 在每一个时钟节拍都要对启动总线 402 测试。一旦发现总线 402 上有启动电平,则马上定义当前的节拍为 H0 节拍。因此,在进入下一个节拍 H1 时,全部 CMPD 部件 411 能同时进入了调用竞争阶段。

在调用竞争阶段, CMPD 部件 411 实现控制,仲裁一个优先权

最高的调用有效。各机的优先权预先写入 R&CD 部件 413 中本机权寄存器 LOCR。调用有效的 CMPD 部件 411 输出信号 414=1。

在时钟节拍控制下，由全部 SNCD 部件 407 中的 CMPD 部件 411 合作，用一个时钟节拍 H1，完成的调用竞争阶段的操作。过程如下：

在 H1 的前半拍，若 CAPD 部件 410 在 H0 送出启动电平，则 CMPD 部件 411 控制将本机权寄存器的内容送到数据总线 404；若有多个 CAPD 部件 410 在 H0 送出启动电平，则数据总线 404 上就会有多个本机权进行“或”操作；在 H1 的后半拍前沿，CMPD 部件 411 控制将数据总线 404 上数据写入暂存器 TMP，并立即和本机权寄存器内容在比较部件 COMP 内进行最高位的比较。比较结果经 415 送 CMPD 部件 411。若最高位相同，表示本 SNCD 部件的调用有效，CMPD 部件 411 的输出线 414=1。

C&ID 部件 412 控制的 Hint 节拍在 H1 节拍之后。C&ID 部件 412 根据线 414 的信息确定序网调用参数的源，由全部 SNCD 部件 407 中的 C&ID 部件 412 合作，用一个时钟节拍 Hint，完成调用参数的广播。过程如下：

在 Hint 的前半拍，确定为源的 C&ID 部件 412 控制调用参数寄存器 VECT 向数据总线 404 送出调用参数；在 Hint 的后半拍前沿，全部 C&ID 部件 412 控制将数据总线 404 上的调用参数写入暂存器 TMP。这是序网调用的调用参数。

暂存器 TMP 中的数据是可以由计算机指令经总线 405 读取的，因此，在 Hint 阶段为各个中断程序准备了序网的调用参数。

在 Hint 的后半节拍，各个 C&ID 部件 412 向计算机发出一个中断电平信号 408，使计算机进入中断程序。发出中断电平和 H1，Hint

阶段的结果 415、414 没有关系。只要启动总线 402 上出现信号，就一定有中断电平会发生。

调用申请的屏蔽。从调用申请开始，各 SNCD 部件进入一个能屏蔽新的序网调用指令的状态。解除屏蔽要在中断程序读取 TMP 中的调用参数之后，从而保证中断程序读取的调用参数的正确性。

多个序网调用指令在同一个节拍发生，进入本机权的竞争，是本机权优先。不同节拍的序网调用指令受到屏蔽处理，是时间优先。调用屏蔽从 H1 开始，也就是说只要时间差一个节拍，就是时间优先的处理方式。

本实施例中的本机权有 16 级，用 16 位寄存器 LOCR 中每个位 (=1) 表示 1 个权。16 个计算机各占一位。在多个本机权同时送到数据总线 404 上时，16 位数据总线上出现了多个位有 1 的状态，表现了参与调用竞争的全部本机权信息。各 SNCD 部件读入这个信息并做比较，就可以判断自己的优先权了。

图 4C 是序网调用部件 SNCD 的工作流程表。序网调用部件 SNCD 的工作流程表是按照时钟节拍的时序 H0、H1、Hint 排列的。表中详细地列出了每一个时钟节拍的工作内容。表中第一行是本文前面叙述过的序网调用的阶段分割。第二行是时钟节拍的名称。第三行的 1, 0 表示了时钟节拍的电平，在表中用 1 表示前半拍，用 0 表示后半拍。第四行是有序网调用的 SNCD 部件，并在调用竞争阶段获得调用有效，其在各个时钟节拍的工作内容。第五行是有序网调用的 SNCD 部件，并在调用竞争阶段竞争失败，其在各个时钟节拍的工作内容。第六行是没有调用启动的 SNCD 部件，其在各个时钟节拍的工作内容。

调用子序网部件在流程表中的操作如下：当调用指令执行，该

SNCD 部件接收来自计算机的序网调用指令。序网调用指令在 SNCD 部件内分解为调用位置和调用参数。调用位置产生调用启动电平。

在 H0 的前半节拍，如表中第 4、5 行所示，有二个 SNCD 部件发出启动电平到启动总线 402，它们在或总线 402 上，会合成一个启动电平，传播到全部 SNCD 部件。

在每一个时钟后半节拍前沿，各 SNCD 部件对启动总线 402 进行测试，一旦测到启动电平，则马上定义该时钟节拍为 H0。下一个节拍为调用竞争阶段 H1。多 SNCD 部件从 H1 开始进入合作。

在 H1 的前半节拍，在 H0 节拍输出启动电平的 SNCD 部件，将 SNCD 部件本机权寄存器的内容送数据总线 404。如表中第 4、5 行所示，有二个 SNCD 部件发出的本机权在数据总线 404 上进行“或”运算。在数据总线 404 上产生多个本机权的和，简称全局的权。

在 H1 后半节拍的前沿，各 SNCD 部件控制，数据总线 404 上全局的权被写入暂寄存器 TMP。然后，在 H0 节拍输出启动电平的 SNCD 部件立即开始进行“暂寄存器 TMP 的最高位和本机权”的比较。比较结果相同，如表中第 4 行所示，表示本 SNCD 部件的调用有效；比较结果不相同，如表中第 5 行所示，表示本 SNCD 部件的调用无效。

在 Hint 节拍，有二件相互独立的工作要完成。一件是发出硬件的中断电平，另一件是调用参数的广播。

在 Hint 的后半拍，由各 SNCD 部件发出中断电平，送到和该 SNCD 部件连接的计算机。

在 Hint 的前半节拍，在 H1 竞争产生有效的 SNCD 部件将调用参数寄存器 VECT 的内容送数据总线 404，实现了调用参数的广播。如表中第 4 行所示，只有一个 SNCD 部件送出数据到数据总线 404。

在 Hint 后半节拍的前沿，各 SNCD 部件控制，数据总线 404 上的内容被写入暂存器 TMP。暂存器 TMP 的内容被保持，直至中断程序来读取暂存器 TMP 中的调用参数

至此，调用子程序网部件的硬件处理过程结束。

图 4D 是 CAPD 部件的合作控制过程。CAPD 部件 410 接受来自计算机的调用指令，并发出下一步合作控制的 H1 信号。图中双线表示有工作信号，并省略了非工作状态的部件模块。

调用申请部件 CAPD410 包括一个地址译码部件 ADDR428 和一个 H0 测试部件 HTST429。图中，左侧的 CAPD 部件 410 接受来自计算机总线 405 的调用指令。指令的调用参数写入 R&CD 部件 413 的调用参数寄存器 VECT。同时，地址译码部件 ADDR428 进行地址译码，并产生启动电平。在下一个节拍开始，启动电平 420 送到启动总线 402。启动电平宽度为一个时钟节拍。启动的信号 420 同时送 CMPD 部件 411。

所有的 SNCD 部件的 H0 测试部件 HTST429 在每一个节拍的后半拍前沿测试启动总线 402。若总线 402 上有启动电平，则立即定义本时钟节拍为 H0，送出下一个节拍的合作时钟 H1 到 CMPD 部件 411。

图 4E 是 CMPD 部件的合作控制过程。CMPD 部件接受来自 CAPD 部件的启动信号 420 和合作时钟信号 H1，进行全局的竞争比较，并在最后送出调用有效的信号 414，和下一个节拍的合作时钟 Hint 到 CMPD 部件 412。图中双线表示有工作信号，并省略了非工

作状态的部件模块。

调用竞争部件 **CMPD411** 包括一个竞争判断部件 **VALD430** 和一个控制信号器 **RCTL431**。

在本 **SNCD** 部件发出启动电平 **420** 的条件下,如图中左侧所示:在 **H1** 的前半拍,控制信号器 **RCTL431** 送出信号 **432** 到本机权寄存器 **LOCR**,控制 **LOCR** 中的本机权经 **434** 送数据总线 **404**。同时,根据 **H1** 信号,全部部件 **RCTL431** 送出信号 **433** 控制暂存器 **TMP**,使在 **H1** 的后半拍前沿, **TMP** 锁定数据总线 **404** 上的数据。

TMP 中的数据是参与竞争的全局的权,在 **H1** 后半拍经 **435** 送 **413** 部件中的比较部件 **COMP**。同时,本机权 **LOCR** 经线 **436** 也送 **413** 部件中的比较部件 **COMP**。比较部件的结果经 **415** 送 **CMPD** 部件 **411** 的竞争判断部件 **VALD430**。竞争判断部件 **VALD430** 读入来自 **R&CD** 部件 **413** 的比较信号 **415**,在下一个节拍产生送 **C&ID** 部件 **412** 的信号 **414**。

根据 **H1** 信号,部件 **RCTL431** 送出下一个节拍的合作时钟 **Hint**。

在有多个 **SNCD** 部件竞争的情况下,各 **SNCD** 部件的操作和图中左侧部件的操作一样,但是,部件 **413** 中的比较部件 **COMP** 送出的信号 **415** 内容不同。

在本 **SNCD** 部件没有发出启动电平 **420** 的条件下,如图中右侧所示:根据 **H1** 信号,部件 **RCTL431** 在 **H1** 节拍送出信号 **433** 控制暂存器 **TMP**,使在 **H1** 的后半拍前沿, **TMP** 锁定数据总线 **404** 上的数据。根据 **H1** 信号,部件 **RCTL431** 送出下一个节拍的合作时钟 **Hint**。

图 4F 是 **C&ID** 部件的合作控制过程。**C&ID** 部件接受来自

CMPD 部件的调用有效信号 **414** 和合作时钟信号 Hint, 进行调用参数的广播和送出中断电平。图中双线表示有工作信号, 并省略了非工作状态的部件模块。

调用参数和中断部件 **412** 包括一个中断电平发生器 **INTT445** 和一个控制信号器 **CTLR446**。中断电平发生器 **INTT445** 在收到 Hint 信号后, 送出中断电平 **408** 到计算机。

控制信号器 **CTLR446** 输出调用参数广播的控制信号。在 Hint 节拍内, 如果输入 $414=1$, 如图中左侧所示, 表示调用有效, 部件 **CTLR446** 送出控制信号 **447** 控制调用参数寄存器 VECT, 使在 Hint 的前半拍, 调用参数寄存器 VECT 经 **449** 送出调用参数到数据总线 **404**; 如果 $414=0$, 如图中右侧所示, 则没有数据送出。

根据 Hint 信号, 所有的部件 **CTLR446** 送出控制信号 **448** 到暂存器 TMP, 使在 Hint 的后半拍前沿, 暂存器 TMP 能够锁定数据总线 **404** 上的数据。这个数据是调用参数, 和序网的入口地址有关。

图 4G 是寄存器和比较部件内的比较部件 COMP 结构。本图描述 R&CD 部件 **413** 中的比较部件 COMP 中, 全局的权(在 TMP 中)和本机权 (LOCR) 进行最高位比较的结构。

在 H1 的后半拍前沿, 暂存器 TMP 锁定了数据总线上的全局的权, 马上进入和本机权 LOCR 的比较。它们最高位相同, 就是调用有效。

暂存器 TMP 的 16 位输出 **439** 送到“留最高位”的部件 **435**。留最高位部件 **435** 的输出 **444** 和来自本机权寄存器的输出 **438**, 一起送入数据比较器 **436**。数据比较器 **436** 的相等输出 **415** 送 CMPD 部件 **411**。

37

保留最高位的电路结构是一个简单的处理电路 435，有 16 位的 D 输入和 Q 输出。如图所示，在 DO、D1、D2、-- D15 的序列中，任何一位（例如 D1）出现 1，会使在该位后的全部位的 Q（例如 Q2-Q15）输出端输出为 0。仅一位（例如 Q1）是 1。

DO、D1、D2、-- D15 的优先权排队，是本实施例的设计，允许有其他不同的优先规则。

COMP 部件中还包括其他的寄存器数据比较模式。由于各种大小，等于的比较器是通用设计，在 CAD 设计软件的标准库中就有。因此，在下面的描述中将不再作说明。

在上述结构和过程说明后，SNCD 部件实现调用子序网的结构和步骤已经清楚，包括每个节拍中寄存器的读写控制，时序的控制和传送路径。进一步的设计就是不同节拍，对各个寄存器读写操作的控制和简单逻辑。一个熟悉 EDA 设计工具的硬件设计工程师应当能够设计图 4A、B、C、D、E、F 所述的 SNCD 部件，一个软件工程师也能完成有关的中断程序。

在上面的发明中描述了调用子序网部件。从图 5 开始，在调用子序网部件的申请类型上，做进一步的发展。

图 5 用于说明序网的级调用和序调用。在调用子序网部件的基础上，本发明进一步将序网调用申请进行分类，创建二个新的序网调用申请形式，它们是序网的级调用和序网的序调用。

级调用申请和序调用申请是在“调用指令--调用启动电平”过程中增加的一个新的条件。条件不满足，不发生序网调用，条件满足，启动序网调用。在启动序网调用时，图 4 所述的 H0，H1，Hint 节拍也要有相应的调整。级调用和序调用的调整是不同的，当然最后所有的逻辑要合并，成为一个 SNCD407 结构。

在序网调用申请阶段的 Happ 节拍，级调用和序调用的操作是在各 SNCD 部件 407 内执行级调用条件和序调用条件的检查。在条件满足的情况下，SNCD 部件 407 才向启动总线 402 发出启动电平。

本发明引入了序网的级调用申请，可以支持处理重要性不同的外部事件和序网的实时响应的能力；本发明引入了序网的序调用申请，可以支持程序处理的有序事件，支持复杂的程序结构。

图 5A 是本发明的序网级调用结构。序网级调用基础是序网的“级属性”概念。其原理是：每一个序网都有级，在级调用申请时申明，申请成功，则申请级进入系统级标志（简称系统级 SYSJ）。申请级调用指令和写申请级指令来自同一个程序。在序网级调用的申请后，系统级 SYSJ 和申请级进行比较。比较原则是：申请级高于系统级 SYSJ，级调用申请许可，产生启动电平；申请级等于或低于系统级 SYSJ，级调用申请不许可，没有启动电平。当级调用的子序网运行结束，系统级标志 SYSJ 中的最高位要退出（清零）。序网的级调用建立了多机系统的级特征。

本图描述了支持序网级调用的结构，也是图 4A，4B 所示的 SNCD 部件的一个实施例。和图 4B 的过程相比，本实施例要增加一个检查级调用申请的环节。由于这个新的环节是本地操作，要在调用启动电平前处理，因此，该环节被安放在第 I 阶段。在第 I 阶段内形成“级调用指令—级申请许可—序网调用启动电平”的步骤关系。其他 II，III，IV，V，VI 阶段的格局保持不变。但是，新加入的申请级参量处理需要增加一个“多机同一个节拍申请，又有相同最高申请级的”竞争，使调用竞争阶段需要 H1、H2 二个节拍。

在结构上，和图 4B 相比，仍保持了四个模块的结构。其中 CAPD 部件 410、CMPD 部件 411、C&RD 部件 412 分别用于调用申请阶段、调用竞争阶段、调用参数和中断阶段的管理，各阶段在时间上

是连接的。但是，模块内部有下列变化：

在 R&CD 部件 413 中增加了系统级 SYSJ 和申请级 PJB 二个寄存器。部件 413 中的比较部件 COMP 内部要增加 PJB 大于 SYSJ 的比较电路结构，比较结果经 501 送 CAPD 部件 410；增加暂存器最高位等于申请级寄存器 PJB 的比较电路结构，比较结果经 502 送 CMPD 部件 411。在 EDA 设计中，这些比较电路有标准模块。

在 CAPD 部件 410 中增加了申请级的许可处理，增加了一个 Happ 节拍。

在 CMPD 部件 411 中形成二个节拍 H1, H2 来处理申请级的调用竞争。

在 C&ID 部件 412 内部的结构不变。

序网的级调用指令是单机指令格式。该指令是计算机的 IO 指令，需要占用 IO 的地址进行级调用参数的输入。级调用指令的参数是“级调用的申请级”和“序网的调用参数”。本实施例中，申请级是先于调用参数写入 SNCD 部件 PJB 寄存器的。在写“调用参数”兼级调用指令时，地址译码使 CAPD 部件 410 进入级申请检查。

增加一个本地的级申请节拍 Happ，进行级许可检查。级许可检查的内容是比较“申请级 PJB”和“系统级 SYSJ”的大小。如果申请级 PJB 高于系统级 SYSJ，产生一个许可信号 501 送 CAPD 部件 410，使在写“序网的调用参数”时，CAPD 部件 410 产生启动电平送启动总线 402。如果申请级等于或低于系统级 SYSJ，许可信号 501=0，则在写“序网的调用参数”时，没有启动电平。

由于级调用的申请是否有效要在启动电平之前确定，系统级 SYSJ 的寄存器必须是分布结构的，并有分布内容的一致性。在本

实施例中，系统级 SYSJ 用位表示，系统级 SYSJ 寄存器有 16 个位，表示系统的 16 个级。系统级 SYSJ 中出现多个“1”是多次序网级调用申请成功的记录。申请级也用一个 16 位的数据表示，每一位表示一个申请级，申请级只有一个位出现 1。系统级 SYSJ 有更新的能力。每当一个序网的级调用成功，新的最高级就要添加在全部 SNCD 部件中的系统级 SYSJ 寄存器上。在一个级调用序网的各分支程序都结束后，系统级 SYSJ 寄存器中的最高级就要退出。

级调用申请成功，启动电平在 H0 节拍传送到全部的 CAPD 部件 410。并在下一个节拍转入调用竞争阶段 H1。

在多机程序同时出现序网的级调用指令，并且申请级都高于系统级 SYSJ 情况下，要在调用竞争阶段，仲裁出一个级调用有效。本发明的调用竞争阶段分二步。第一步在 H1 节拍比较申请级高低。由全部申请级调用 CMPD 部件合作控制完成信息传送，在 R&CD 部件 413 中比较多个级调用的申请级，最高级经 502 送 CMPD 部件；第二步在 H2 节拍比较有最高申请级的各 SNCD 部件的本机权。由在 H1 节拍中的获得最高级的 CMPD 部件合作控制完成信息传送，在 R&CD 部件 413 中比较本机权，结果经 415 送 CMPD 部件 411。本机权在 LOCR 寄存器中。

调用竞争阶段结束后进入调用参数和中断阶段，产生调用参数的广播和硬件的中断信号，系统进入序网。

图 5B 是本发明序网级调用工作流程表。序网调用部件 SNCD 的工作流程表是按照时钟节拍的时序 H_{app}、H₀、H₁、H₂、H_{int} 的排列，详细地列出了每一个时间节拍的工作内容。表中第一行是本文前面叙述过的序网调用的阶段分割。第二行是时钟节拍的名称。第三行的 1，0 表示了时钟节拍的电平，在表中用 1 表示前半拍，用 0 表示后半拍。第四行是有级调用申请的 SNCD 部件，在申请阶

段获得申请有效，在调用竞争阶段获得调用有效，其在各个时钟节拍的工作内容。第五行是有级调用申请的 SNCD 部件，在申请阶段获得申请有效，在调用竞争阶段因本机权竞争失败，其在各个时钟节拍的工作内容。第六行是有级调用申请的 SNCD 部件，在申请阶段获得申请有效，在调用竞争阶段因申请级竞争失败，其在各个时钟节拍的工作内容。第七行是有级调用申请的 SNCD 部件，因申请级低于运行的级，在申请阶段失败，其在各个时钟节拍的工作内容。第八行是没有调用申请的 SNCD 部件，其在各个时钟节拍的工作内容。

调用子序网部件在流程表中的操作如下：SNCD 部件接收来自计算机的级调用指令，经地址译码识别，在下一个 Happ 节拍进行申请级（已经先于级调用指令写入了 SNCD 部件的申请级寄存器）和系统级 SYSJ 的最高位进行比较。若申请级低于或等于当前的系统级 SYSJ，则级调用申请不被接受。若申请级高于当前的系统级 SYSJ，则级调用申请被接受，该 SNCD 部件在下一个时钟节拍 H0 发出启动电平。如表中所示，有四个 SNCD 部件出现了级调用指令，只有三个 SNCD 部件级调用申请有效。

在 H0 的前半节拍，启动总线 402 上有启动电平出现。如表中所示，有三个 SNCD 部件都发出了启动电平。它们在 H0 节拍，在或总线 402 上，会合成一个，传播到全部 SNCD 部件。

在每一个时钟的后半节拍前沿，各 SNCD 部件测试启动总线 402。一旦测到启动电平，则马上定义此时钟节拍为 H0，并在下一个节拍进入调用竞争阶段 H1。

在 H1 的前半节拍，在 H0 节拍输出启动电平的 SNCD 部件，将 SNCD 部件内部的申请级寄存器 PJB 的内容送数据总线 404。数据总线 404 进行或运算。如表所示，有三个 SNCD 部件发出的申请

级信息在数据总线 404 上进行或运算。

在 H1 后半节拍的前沿，各 SNCD 部件将数据总线 404 上的内容写入暂存器 TMP，然后，在 H0 节拍输出启动电平 of SNCD 部件立即开始进行“暂存器 TMP 的最高位和申请级”进行比较。比较结果相同，表示本 SNCD 部件的调用有效。本表中的第 4，5 行有相同的最高申请级，在 H1 节拍的竞争中调用有效。

在 H2 的前半节拍前沿，暂存器 TMP（内容为全部级调用的申请级）中的最高位，写入到 SNCD 部件中的系统级 SYSJ 寄存器（系统级 SYSJ 寄存器中其余的位保持不变）。记录系统新的最高级。

在 H2 的前半节拍，在 H1 节拍中调用有效的 SNCD 部件将本机权寄存器的内容送数据总线 404。数据总线 404 是或总线，如图所示，有二个 SNCD 部件发出的本机权信息在数据总线 404 上做或运算。

在 H2 后半节拍的前沿，各 SNCD 部件将数据总线 404 上的内容写入暂存器 TMP，然后，在 H1 节拍后调用有效的 SNCD 部件立即开始进行“暂存器 TMP 的最高位和本机权”进行比较。比较结果相同，表示本 SNCD 部件的调用有效。本表中的第 4 行表示在 H2 节拍竞争中调用有效。在决出调用有效的 SNCD 部件后，进入由 C&ID 部件处理的调用参数广播和中断阶段 Hint。

在 Hint 节拍，有二件相互独立的工作要完成。一件是发出硬件的中断电平，另一件是调用参数的广播。

在 Hint 的后半拍，各 SNCD 部件发出中断电平，送和其连接的计算机。

在 Hint 的前半节拍，从 H2 节拍后调用有效的 SNCD 部件（本

43

表中的第 4 行) 将调用参数寄存器 VECT 的内容送数据总线 404, 实现调用参数广播。只有一个 SNCD 部件能送出数据到数据总线 404。

在 Hint 后半节拍的前沿, 各 SNCD 部件将数据总线 404 上的内容写入暂寄存器 TMP。暂寄存器 TMP 的内容被保持, 直至中断程序读取了暂寄存器 TMP 中的数据。

至此, 序网级调用的硬件处理过程结束。

在序网的全部分支程序结束时, 系统级 SYSJ 寄存器中的最高级退出。具体的硬件设计可以有很多种。例如, 在序网的第 N+1 个程序中, 设计一条 SYSJ 最高级退出的新指令, 该指令运行时, 级调用退出信息经各 SNCD 部件的解释, 由硬件同步地来擦除分布的系统级寄存器 SYSJ 的最高位。

在上述流程表的过程说明和内容说明后, SNCD 部件实现序网级调用的功能结构和步骤已经清楚, 包括在每个节拍中寄存器的读写控制, 时序的控制和路径控制。另外一个新的内容是 COMP 中的各类比较器, 它们是有标准库的。还有一个就是 SYSJ 的最高位的写和最高位的清除, 找到最高位的电路在图 4G 中已经有描述。因此, 进一步的设计就是不同节拍, 对各个寄存器读写操作的控制和一些简单逻辑。一个熟悉 EDA 设计工具的工程师应当能够设计图 5A、B 所述的序网的级调用结构和 SNCD 部件, 一个软件工程师也能完成有关的中断程序。

图 5C 是本发明的序网序调用结构。序网序调用是同一个级属性下, 建立多序网之间的执行顺序。其原理是: 全局指定下一个序调用的许可机号 (存 NEXT 寄存器), 在序调用申请时检查, 申请成功, 该许可撤消。由于可能同时产生多个序调用指令, 本发明通

过指定一个序调用许可，来处理调用竞争。和级调用不同，申请序调用指令和指定下一个序调用许可的指令不是来自同一个程序。

本实施例在每一个 R&CD 部件 413 中，建立指示下一个序调用机号的 NEXT 寄存器。当 SNCD 部件接收到一个序调用申请（写调用参数）时，NEXT 寄存器和本机的机号 LOCN 进行比较。比较结果相同，产生调用启动电平，表示系统将接受该序调用申请。否则调用申请无效。

先有的序网计算机发明中，第 $N+1$ 个程序 P_s 是一个全局的程序，由它来处理 NEXT 寄存器的修改，可以使序调用许可的机号同时写入分布的 NEXT 寄存器，保证分布结构的 R&CD 部件 413 中的 NEXT 寄存器的一致性。一个序调用申请成功，NEXT 寄存器中的内容就无效。一个 NEXT 寄存器写操作可以激活一次序调用许可。

利用 P_s 程序来并行地写全部 NEXT 寄存器中的数据，在先有的序网计算机发明中通过 CECO 指令的 IO 写就能实现。本实施例的描述是在 NEXT 参数已经写入以后，序网序调用的结构和过程。

图 5C 描述了支持序网序调用的结构，也是图 4A，4B 所示的 SNCD 部件的一个实施例。和图 4B 的过程相比，本实施例要增加一个处理序调用申请的环节。由于这个新的环节是本地操作，要在调用启动前处理，因此，该环节被安放在第 I 阶段。在第 I 阶段内形成“序调用指令—序申请许可—序网调用启动”的步骤关系。其他 II，III，IV，V，VI 阶段的格局保持不变，但是，新加入的序调用许可参量要求在调用竞争阶段有一个判别是否是序调用的节拍。

在结构上，和图 4B 相比，仍保持了四个模块的结构，其中 CAPD 部件 410、CMPD 部件 411、C&RD 部件 412 分别用于调用申请阶

段、调用竞争阶段、调用参数和中断阶段的管理，各阶段在时间上是连接的。但是，也有下列变化：

在 R&CD 部件 413 中增加了 NEXT 机号和本机号二个寄存器。部件 413 中的 COMP 部件还要增加 NEXT 机号和本机号进行等于比较的结构，比较结果经 503 送 CAPD 部件 410，以及增加一个判别 TMP 寄存器全 0 的电路结构，比较结果经 504 送 CMPD 部件 411。504=1 表示是序调用。多序调用之间的竞争在写 NEXT 时已经处理完毕。在 EDA 设计中，这些比较电路有标准模块。

在 CAPD 部件 410 中增加了序调用的许可处理，增加了一个 Happ 节拍。

在 CMPD 部件 411 中增加了判断序调用类型的节拍，用于区别级调用类型。

在 C&ID 部件 412 内部，结构不变。

序网的序调用指令是单机指令格式。该指令是计算机的 IO 指令，需要占用 IO 的地址进行序调用参数的输入。序调用指令的参数是“NEXT”序调用机号和“序网的调用参数”。本实施例中，在写“调用参数”兼序调用指令时，地址译码使 CAPD 部件 410 进入序调用申请过程。

增加一个本地的序申请节拍 Happ，进行序许可检查。序许可检查的内容是比较“NEXT 机号”和“本机号”。如果相同，则产生一个许可信号 503 送 CAPD 部件 410。CAPD 部件 410 产生启动电平送启动总线 402。如果不相同，则没有许可信号，也没有启动电平。

NEXT 寄存器分布在各 SNCD 部件，必须保持内容的一致性。

在本实施例中，机号用位表示，本机号寄存器有 16 个位，表示 16 个机号。NEXT 机号也用一个 16 位的数据表示，数据位上的 1 表示许可机号，NEXT 寄存器中的数值只准有一个位出现 1。

每当一个序调用成功，全部 SNCD 部件中的 NEXT 寄存器清 0。然后由先有的序网计算机发明专利中描述的第 N+1 个程序来写新的 NEXT，指定下一个序调用许可的机号。在出现多个序调用时，由于 NEXT 只有一个，因此，在申请阶段 Happ 节拍，就完成了序调用的竞争。

在 H0 节拍结束后，系统进入调用竞争节拍 H1，H2。

在调用竞争阶段的 H1 前半拍，由 CMPD 部件 411 控制，各 SNCD 部件送入数据总线 404 全为 0。在 H1 后半拍的前沿，0 写入了 R&CD 部件 413 的寄存器 TMP。在 R&CD 部件 413 内部对 TMP 内容进行全 0 判别，结果经线 504 送 CMPD 部件 411。若 504=1（表示 TMP 全 0），由 CMPD 部件确定为序调用。

H2 为空操作。

调用竞争阶段结束后，CMPD 部件送出 414 信号，进入调用参数和中断阶段，产生调用参数的广播和硬件的中断信号，系统进入序网运行。

图 5D 是本发明序网序调用工作流程表。序网调用部件 SNCD 的工作流程表是按照时钟节拍的时序 Happ、H0、H1、H2、Hint 的排列，详细地列出了每一个时间节拍的工作内容。表中第一行是本文前面叙述过的序网调用的阶段分割。第二行是时钟节拍的名称。第三行的 1，0 表示了时钟节拍的电平，在表中用 1 表示前半拍，用 0 表示后半拍。第四行是有序调用申请的 SNCD 部件，在申请竞争阶段获得申请有效，其在各个时钟节拍的工作内容。第五行是有序调用申请的 SNCD 部件，在申请阶段因本机号和 NEXT 机号不同

而失败，其在各个时钟节拍的工作内容。第六行是没有调用申请的 SNCD 部件，其在各个时钟节拍的工作内容。

调用子序网部件在流程表中的操作如下：SNCD 部件接收来自计算机的序调用申请指令。在下一个 Happ 节拍进行 NEXT 序调用机号和本机机号的比较。若本机号和 NEXT 相等，序调用申请被许可。SNAD 部件在下一个时钟节拍 H0 送出启动电平。若本机号和 NEXT 不同，或 NEXT 内容无效，则 SNCD 部件的序调用申请无效。如表中的第 4、5 行所示，有二个 SNCD 部件发出了序调用申请，只有一个通过。

在 H0 的前半节拍，调启动总线 402 上有启动电平，如表所示，只有一个 SNCD 部件发出了启动电平，在 H0 节拍，经过总线 402，传播到全部 SNCD 部件。

在每一个时钟的后半节拍前沿，各 CAPD 部件 410 对启动总线 402 测试，一旦测到有启动电平，则马上定义此时钟节拍为 H0，并在下一个节拍进入调用竞争阶段 H1。

在 H1 的前半节拍，在 H0 节拍向启动总线 402 上输出电平的 SNCD 部件送 0 到数据总线 404。在本实施例结构中，该 0 数据表示序调用。

在 H1 后半节拍的前沿，全部 SNCD 部件将数据总线 404 上的内容写入暂存器 TMP。然后，对暂存器 TMP 的内容进行全 0 检查。检查结果是全 0，表示本次调用是序调用。本表中的第 4 行表示了序调用有效。

在 H1 判断出是序调用后，在 H2 节拍是空操作。由于序调用在申请阶段就只允许一个能进入，因此，在 H0 节拍发出启动电平的 SNCD 部件就是序调用有效的 SNCD 部件。显然，在 H1 节拍的序

调用判别是为了区分级调用。

在 Hint 节拍，有二件相互独立的工作要完成。一件是发出硬件的中断电平，另一件是调用参数的广播。

在 Hint 的后半拍，各 SNCD 部件发出中断电平，送和其连接的计算机。

在 Hint 的前半节拍，从调用有效的 SNCD 部件（本表中的第 4 行）将调用参数寄存器 VECT 的内容送数据总线 404，实现调用参数广播。只有一个 SNCD 部件能送出数据到数据总线 404。

在 Hint 后半节拍的前沿，各 SNCD 部件将数据总线 404 上的内容写入暂存器 TMP。暂存器 TMP 的内容被保持，直至中断程序读取了暂存器 TMP 中的数据。

至此，序网的序调用的硬件处理过程结束。

在上述流程表的过程说明和内容说明后，SNCD 部件实现序网序调用的调用子序网部件的结构和步骤已经清楚，包括在每个节拍中寄存器的读写控制，时序的控制和路径控制。另外一个新的内容是 COMP 中的各类比较器，它们是有标准库的。因此，进一步的设计就是不同节拍，对各个寄存器读写操作的控制和一些简单逻辑。一个熟悉 EDA 设计工具的工程师应当能够设计图 5C、D 所述的序网的序调用结构和 SNCD 部件，一个软件工程师也能完成有关的中断程序。

级调用和序调用在结构上的合并是必然的，这是一个纯粹的逻辑组合问题，因此，本实施例仅作简单的说明如下：

如果级调用和序调用同时发生，在启动总线 402 上合并成一个

49

启动电平。由于序调用的竞争在发出启动电平前已经完成，因此，只需要在调用竞争阶段 H1、H2 完成区分级调用和序调用即可，而在 Hint 节拍，又是同一个结构。

由于级调用在 H1 的前半拍，将本机的申请级 VECT 送数据总线 404，而序调用送 0。因此，若在 H1 后半拍前沿，各暂存器 TMP 锁定的数据为全 0，表示是序调用。反之，是级调用。对于序调用，调用有效的 SNCD 部件已经明确；对于级调用，要继续通过调用竞争阶段，得到调用有效的 SNCD 部件。最后，它们都进入 Hint 节拍处理。

在级调用和序调用同时出现时，上述合并级调用和序调用的结构表明级调用优先，也表明序调用竞争是在同一级别的多个序调用之间进行，不同级别的序网之间没有序调用关系。

多机的调用子序网部件在功能和用途上将超出了单机的调用子程序部件。本发明的调用子序网部件是序网计算机发展通用性的关键技术之一。

50

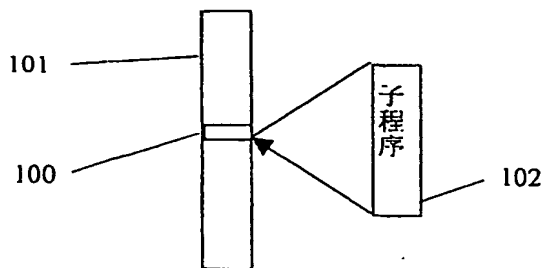


图 1A

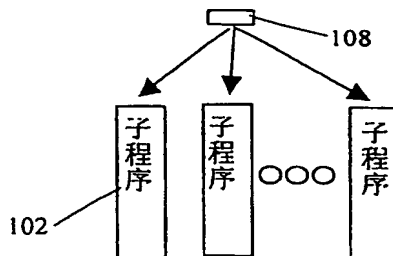


图 1B

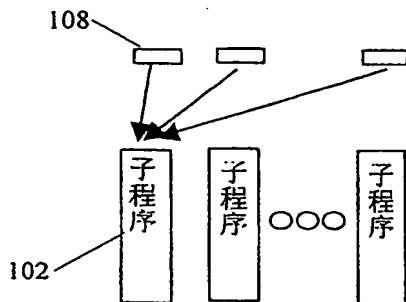


图 1C

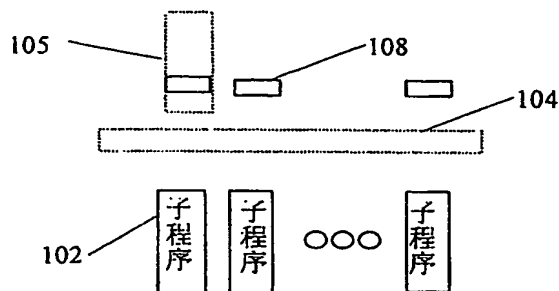


图 1D

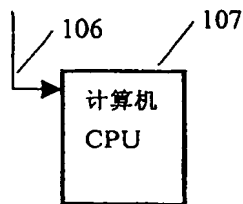


图 1E

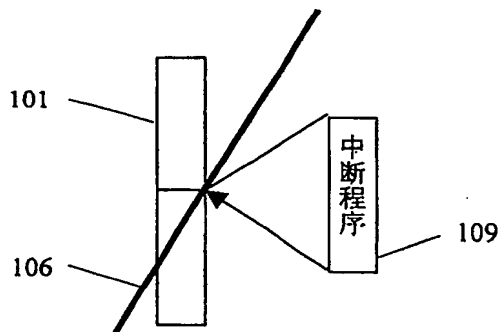


图 1F

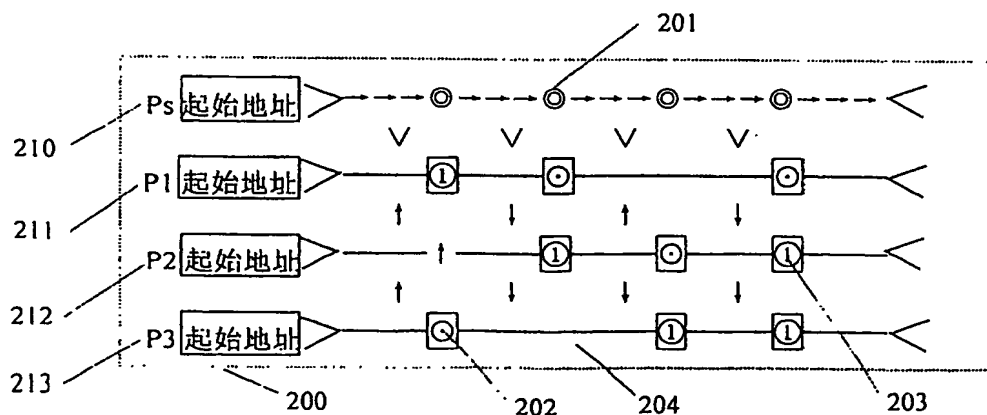


图 2A

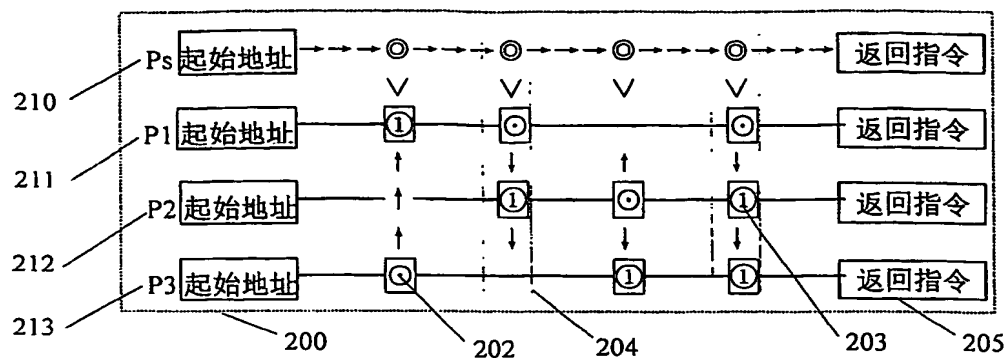


图 2B

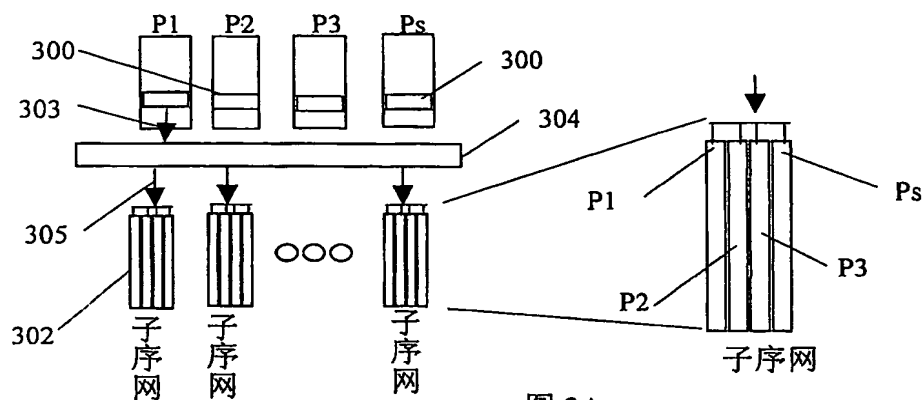


图 3A

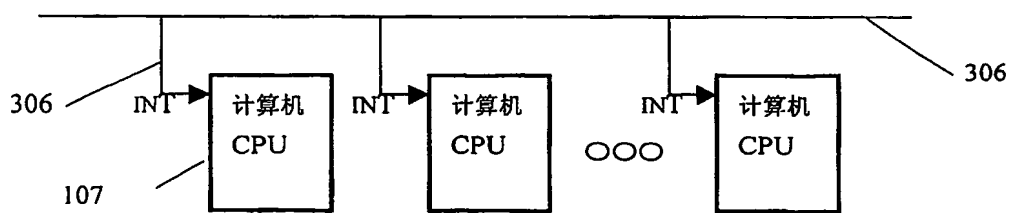


图 3B

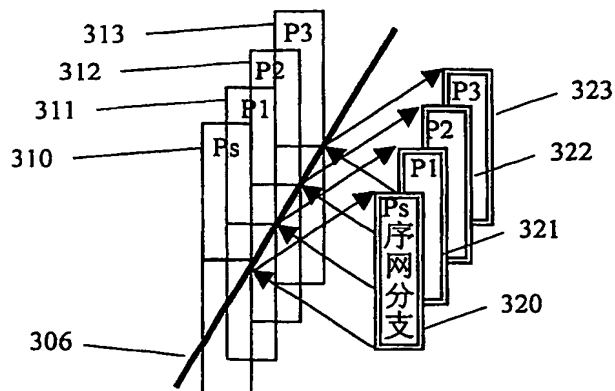


图 3C

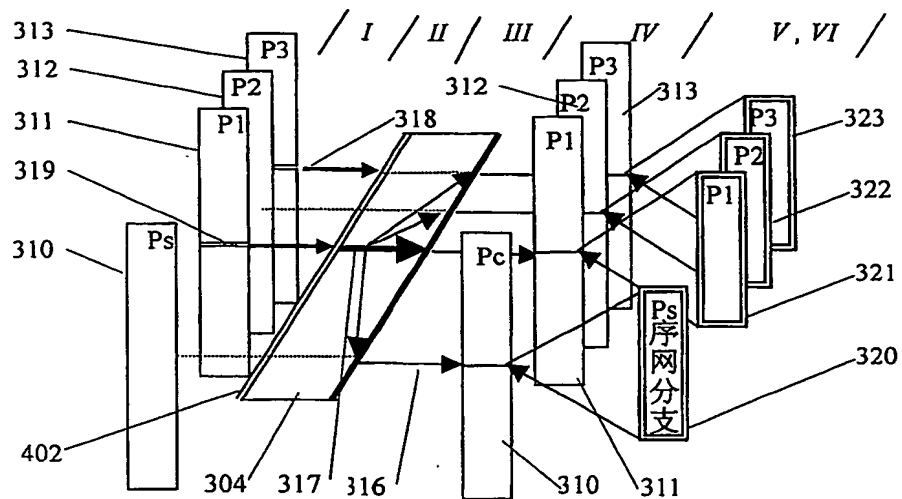


图 3D

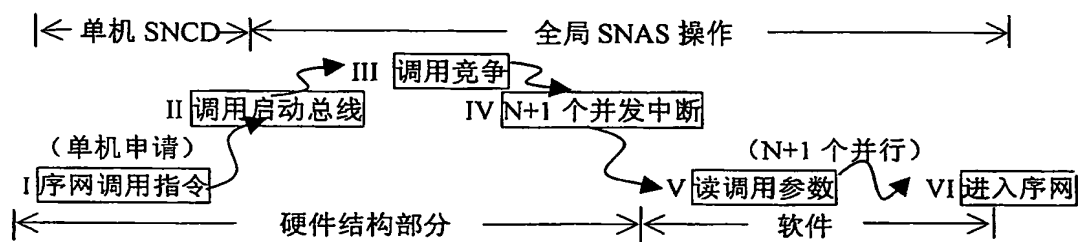


图 3E

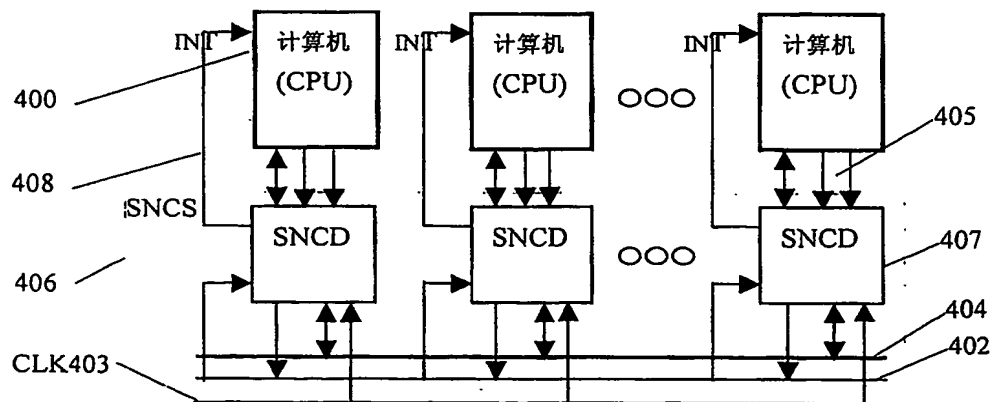


图 4A

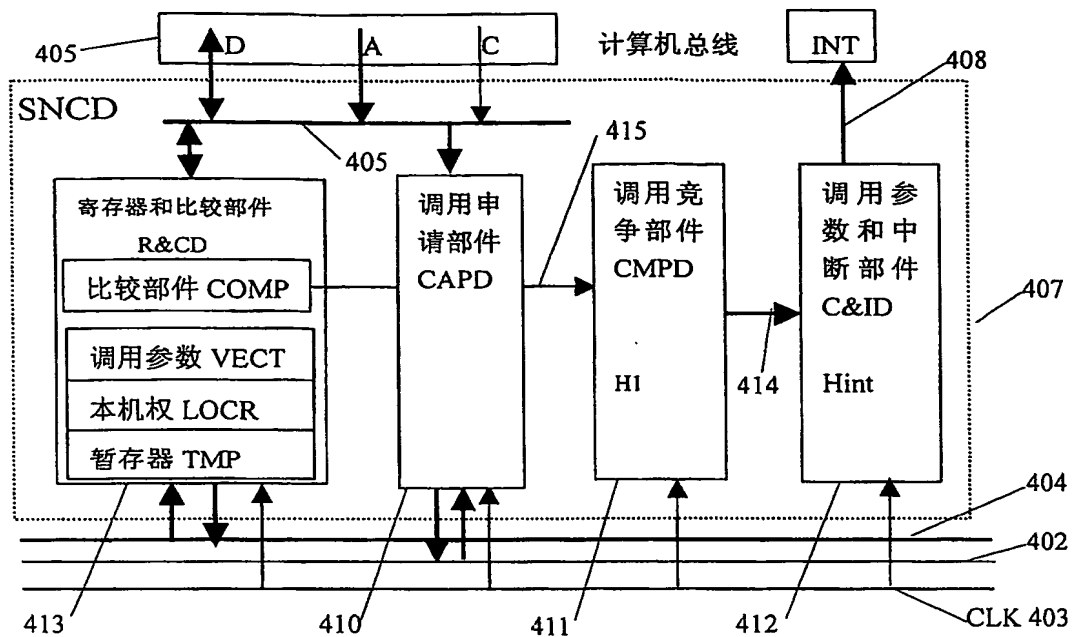


图 4B

序网的调用中，各单元 SNCD 在 H 节拍下的操作内容表

I	II		III		IV	
	H0		H1		Hint	
	1	0	1	0	1	0
SNCD 接收序网调用指令，是申请机	送启动电平到总线 402	前沿，测上全局启动电平。	送本机权到数据总线	前沿，从数据总线读入全局的权，进暂存器。并和本机权比较。最高权相同=调用有效	本 SNCD 调用有效，送调用参数到数据总线	前沿，读数据总线，进暂存器。前沿，发出中断电平到和 SNCD 连接的计算机
同上	同上	同上	送本机权到数据总线	前沿，从数据总线读入全局的权，进暂存器。并和本机权比较。本机权低=调用无效	本 SNCD 调用无效，0000 送到数据总线	同上
SNCD 非申请机	无操作	同上	0000 送到数据总线	前沿，从数据总线读入全局的权，进暂存器。本机无调用申请，不作比较。	0000 送到数据总线	同上

图 4C

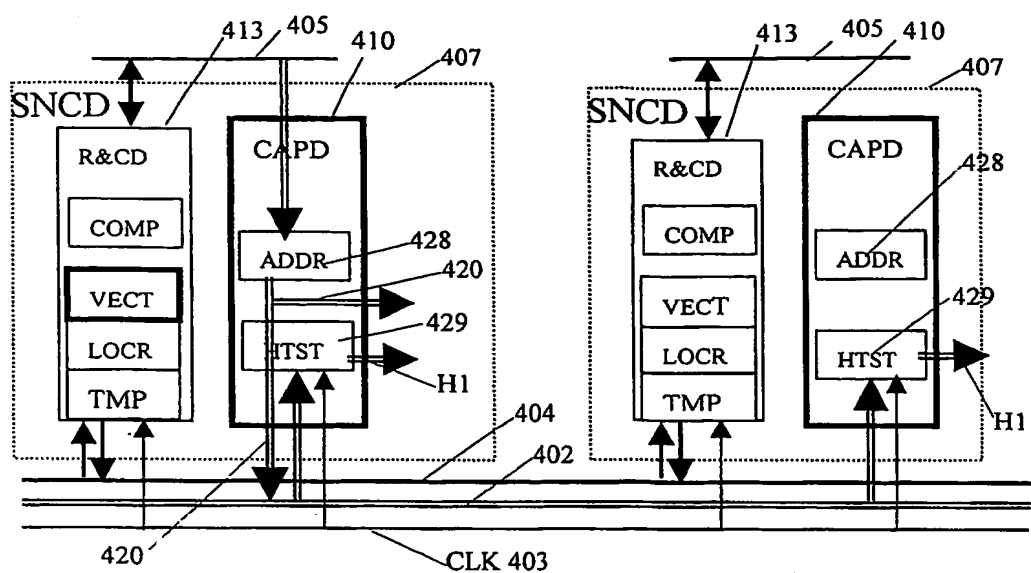


图 4D

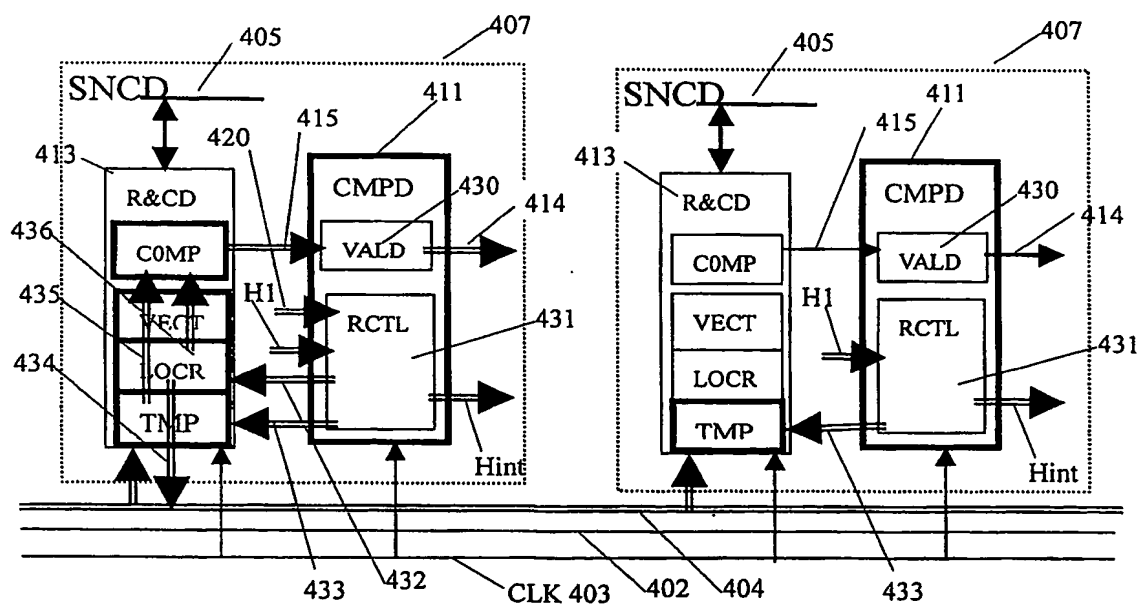


图 4E

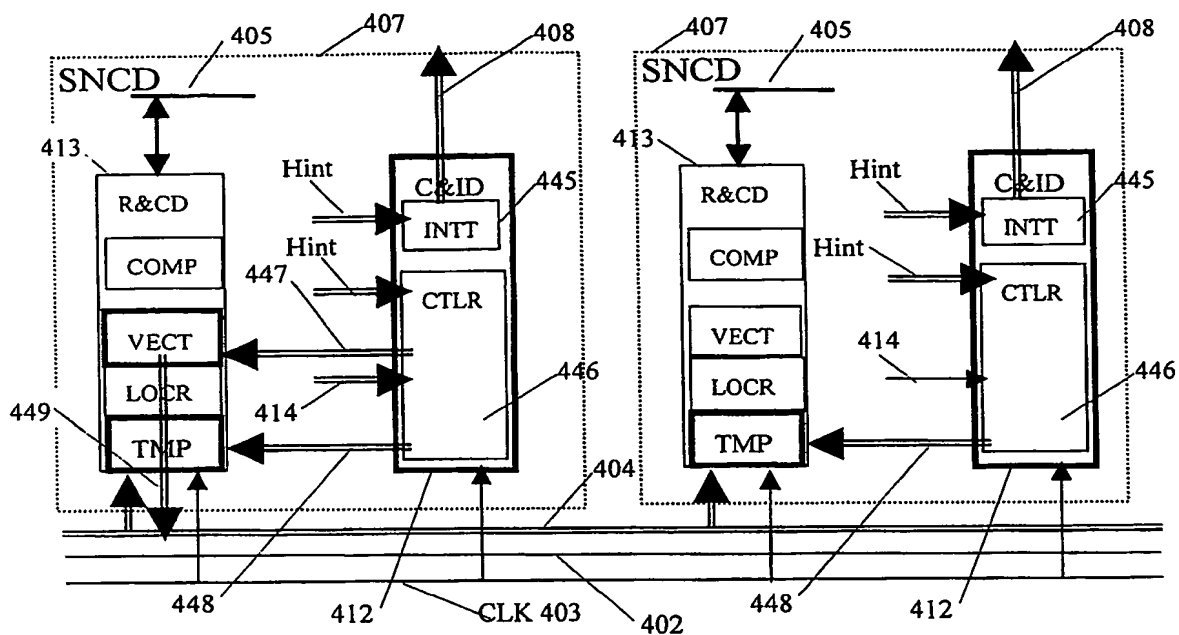


图 4F

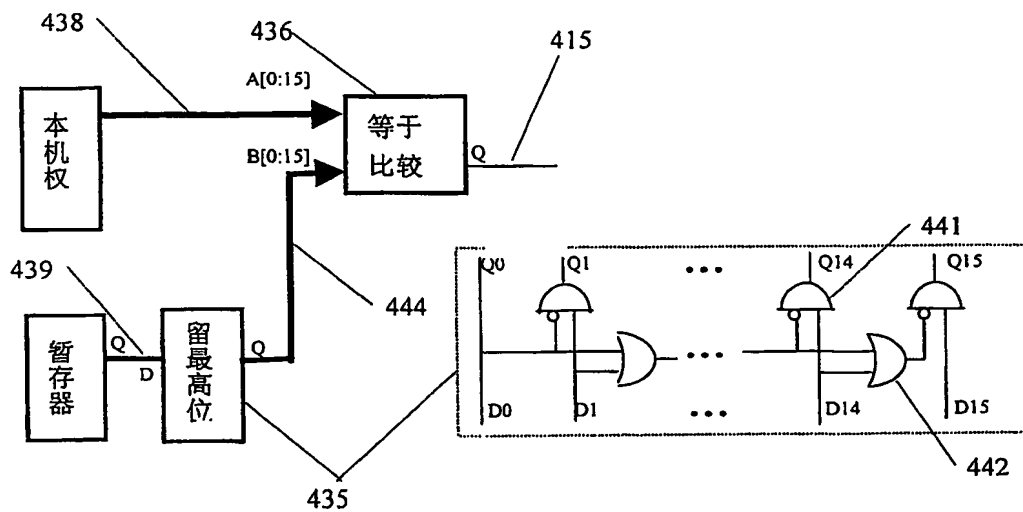


图 4G

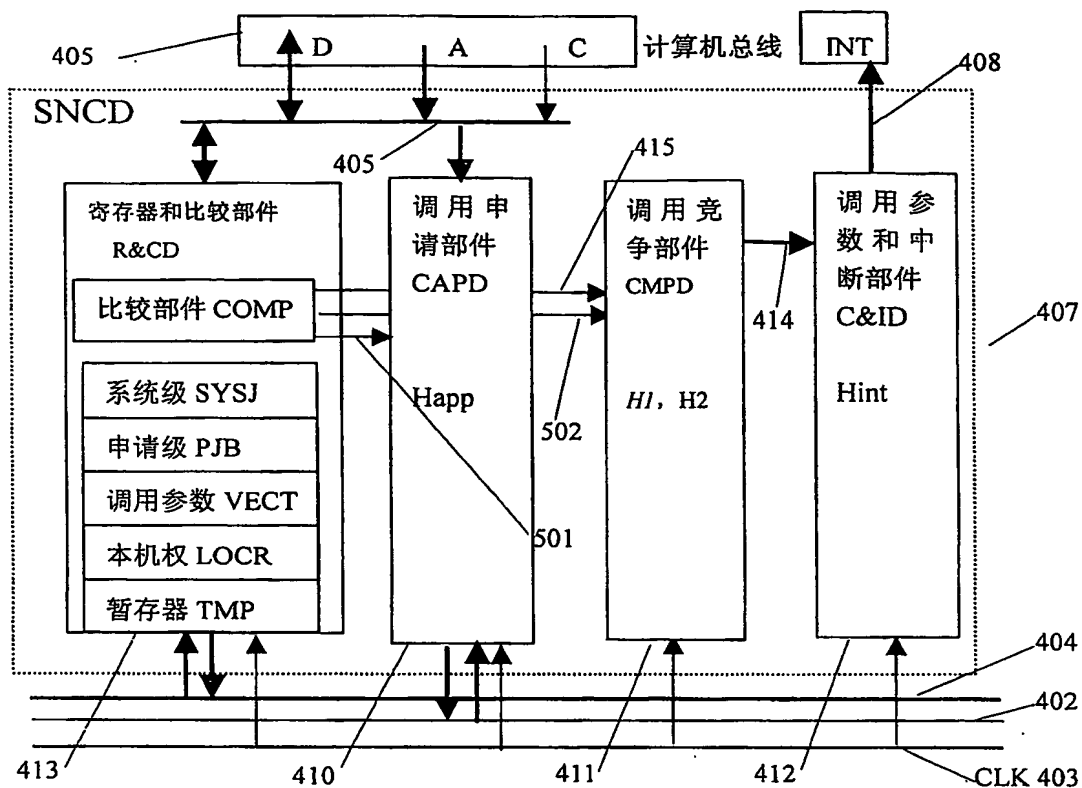


图 5A

序网的级调用中，各单元 SNCD 在 H 节拍下的操作内容表

I		II		III				IV	
Happ		H0		H1		H2		Hint	
1	0	1	0	1	0	1	0	1	0
SNC D 接 收序 网级 调用 指令， 是申 请机	申 请 级 大 于 系 统 级 申 请 有 效	启 动 电 平 送 总 线 402	前 沿 测 402 上 启 动 电 平	送 申 请 级 到 数 据 总 线	前 沿，从数据 总线读入全局 的申请级，进 暂存器。并和 本机申请级比 较。 最高级相同= 调用有效	送本 机 权 到 数 据 总 线。 暂存器中的最 高级写入系统 的级寄存器。	前 沿，从数据总 线读入全局的 权，进暂存器， 并和本机权比 较。 最高权相同= 调用有效	SNC D 有 效， 送调 用参 数到 数据 总线	前 沿， 读数 据总 线， 进暂 存器。 前 沿， 发出 中断 电平 到计 算机
同上	申 请 级 大 于 系 统 级 申 请 有 效	同上	同上	送 申 请 级 到 数 据 总 线	前 沿，从数据 总线读入全局 的申请级，进 暂存器。并和 本机申请级比 较。 最高级相同= 调用有效	送本 机 权 到 数 据 总 线。 暂存器中的最 高级写入系统 的级寄存器。	前 沿，从数据总 线读入全局的 权，进暂存器。 并和本机权比 较。 本机权低= 调用无效	0000 送到 数据 总线	同上
同上	申 请 级 大 于 系 统 级 申 请 有 效	同上	同上	送 申 请 级 到 数 据 总 线	前 沿，从数据 总线读入全局 的申请级，进 暂存器。并和 本机申请级比 较。 最高级不相同 =调用无效	0000 送 数据总 线，暂 存器中 的最 高级 写系 统的 级寄 存器	前 沿，从数据总 线读入全局的 权，进 暂存器。 已调用无 效，不再 作比较。	同上	同上
同上	申 请 级 小 于 系 统 级 申 请 无 效	无 操 作	同上	0000 送到 数据 总线	前 沿，从数据 总线读入全局 的申请级，进 暂存器。	同上	同上	同上	同上
SNC D 非 申请 机	无 操 作	同上	同上	同上	同上	同上	同上	同上	同上

图 5B

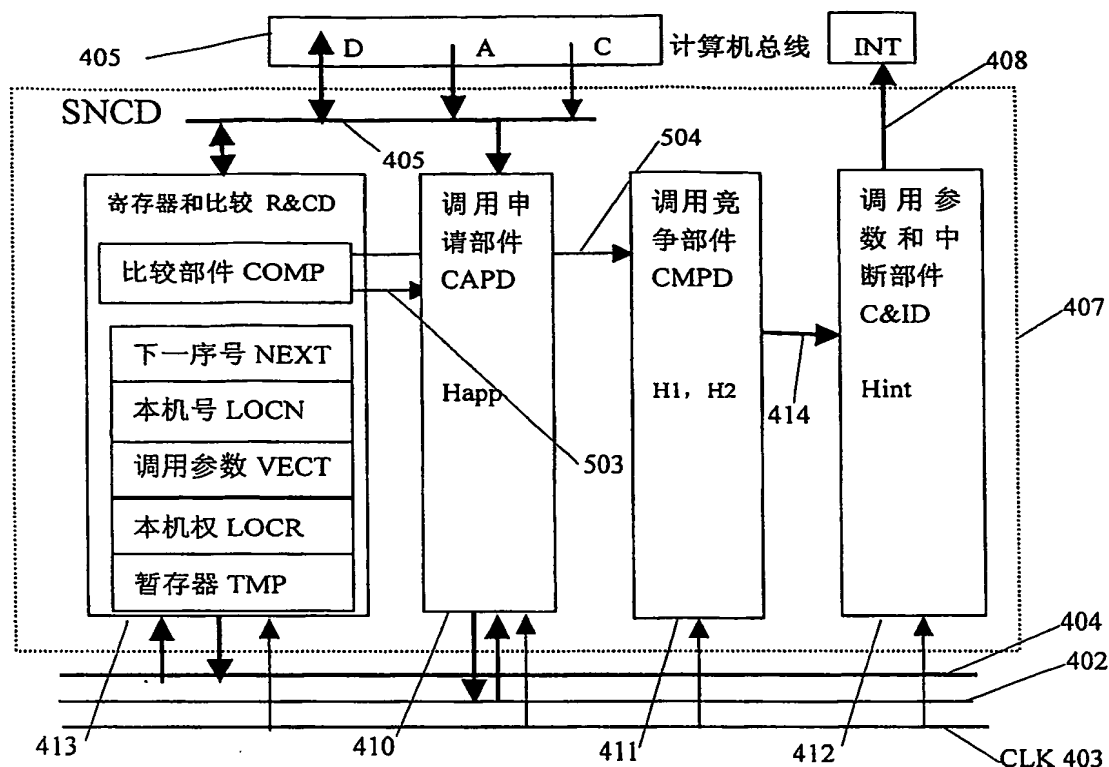


图 5C

序网的序调用中，各单元 SNCD 在 H 节拍下的操作内容表

I			II		III				IV	
Happ			H0		H1		H2		Hint	
	1	0	1	0	1	0	1	0	1	0
SNCD 接收序网序调用指令，是申请机		本机号等于 NEXT 机号，调用有效	送启动电平到总线 402	前沿，测 402 上全局启动电平	是序调用 0000 送到数据总线	前沿，从数据总线读入全局的申请特征，进暂存器。暂存器全 0 是序调用类型。	空操作	空操作	SNC D 有效，送调用参数到数据总线	前沿，读数据总线，进暂存器。前沿，发出中断电平到和 SNCD 连接的计算机
SNCD 接收序网序调用指令，是申请机		本机号不等于 NEXT 机号，调用无效	无操作	同上	无序调用 0000 送到数据总线	同上	空操作	空操作	SNC D 无效，0000 送到数据总线	同上
SNCD 非申请机		无操作	同上	同上	同上	同上	空操作	空操作	同上	同上

图 5D